Reference Document " RBRN-5NETNQ " revision " 1 in database " LCC " status " **uncontrolled - draft** "

Shark Bus Communications version 5

Document Header

Document Content - Signed by Robert Brown/NZ/IVCR on 05/11/2004 12:58:07 p.m., according to /IVCR

VERSION 5.0 (Working)

The Shark Bus is a two-wire (plus external ground), differential signalling, half-duplex communication link. It is designed primarily for only two modules: the Shark Power Module (SPM) and Shark Remote (SR): Generic (SGR) or Invacare (SIR). The electrical specification of the bus does allow for other modules to be connected, and the communication protocol also has spare capacity for other modules. This may be used in future for (say) an Attendant Control Unit or some such peripheral.

CHANGES FROM Version 4.1

1. Removed section 0, which by now is historical.
2. Removed destinations from names of packets. Packets are sent without specifying a receiver, and some packets are intended to be received by multiple modules.
3. Added data from SR and SPM to handle lighting and actuator functionality.
4. Added data from SP to handle SACU.
5. Added packet types 8, 9, and A, for the SACU.
6. Made language more formal in many places
7. Changed bus utilisation examples
8. Added SR translation of HHP data (as has been implemented since version 1.0 of the SR)
9. Added module type to the factory test packet, so that it may also be used for the SACU.
10. Clarified expected behaviour with packets that may be of variable length
11. Added SPM programmable settings packet.
12. Added "Beep on keypress" and "Speed Button Sensitivity" to the SPM programmable settings packet
13. Added "Joystick Switch Threshold" to SPM programmable settings packet.
14. Added "System is driving" to the SPM general packet.

1. Electrical Specification

The two wires are named SBH and SBL ("Shark Bus high" and "Shark Bus low").

The SBH wire is pulled down to zero volts by 270 ohms in parallel with 22k ohms at the SPM and the SR (The 270 ohm terminations switch off when the bus voltages rise above 8V). This is the "recessive" state and it represents a 1. To transmit, modules pull the wire to 5V. This is the "dominant" state and represents a 0. Note that the transmitting circuit will not achieve 5V; 4V is more likely.

The SBH wire is also used by the SR to "wake up" the SPM by asserting more than 12V (actually, it asserts full battery voltage less about a volt) on SBH. The driver for this function in the SR is required to drive the 270 ohm resistors at the SR and SPM ends.
Notes about the terminators:

- In theory, the 270 ohm terminator at the SPM end will be disabled (as it is when the SPM is powered down) but there are other circumstances where the terminator will be turned on.
- The terminator enabling transistor in the SPM may be turned on unpredictably during the power-up period.
- The SR shall disable its own terminator when asserting the wakeup signal.

The wakeup signal on the SBH wire shall be asserted for 300ms +/- 20ms. During this time, the SPM is required to power up, stabilise its operation and latch its power on so that it does not power down again when the wakeup signal is removed. Note that communication on the Shark Bus is not possible when wakeup signal is present. The wakeup signal transmitter shall be designed so that it can source 40mA at greater than 12V, under all normal battery conditions (see Hardware Specification), for the full 300ms.

The SBL wire is pulled up to 5V with a 270 ohm resistor, through a diode. This is the recessive state and it represents a 1. In the dominant state, the SBL wire is pulled down to 0V, and this represents a 0. Note that the transmitting circuit will not achieve 0V; 1V is more likely.

Of course, the recessive and dominant states are simultaneous on both wires.

To summarise:
The recessive state is typically represented by SBH = 0V, SBL = 5V. The recessive noise margin is 5V, and impedance (oversimplified) is 135 ohms.
The dominant state is typically represented by SBH = 4V, SBL = 1V. The dominant noise margin is 3V, and impedance (oversimplified) is 15 ohms.

It is expected that receivers will implement some sort of hysteresis to prevent noisy transitions.
One of the issues to be dealt with by the Bus is ground shift. There are no high-current peripherals that may be connected to the Bus, so the only high-current situation is when a battery charger is connected at the SR. This will cause the SR's B- to shift negative with respect to the SPM. The maximum allowable shift on the B- wire is 1V: when this offset is present, the following conditions will exist:
Recessive state: SBH is pulled to 0V at SPM, -1V at SR. Net effect is -0.5V seen from SPM, 0.5V seen from SR. SBL is pulled to 5V at SPM, 4V at SR. Net effect is 4.5V seen at SPM, 5.5V seen at SR. Noise margin is still 5V at both modules.
SPM asserting dominant: SBH is pulled to 4V (5V seen at SR). SBL is pulled to 1V (2V seen at SR). Noise margin is still 3V.
SR asserting dominant: SBH is pulled to 4V (3V seen at SPM). SBL is pulled to 1V (0V seen at SPM). Noise margin is still 3V.

2. Data format

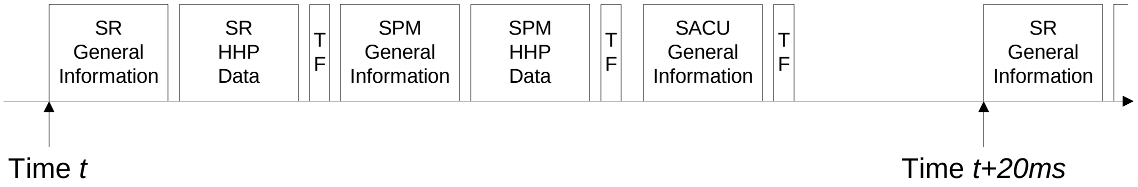
The data shall be transmitted in standard asynchronous serial format as supported by the UARTs on microcontrollers. The data rate shall be 38400bps, with 1 start bit, 8 data bits, and 2 stop bits. With updates every 20ms, this allows for 69 bytes to be carried on the bus for each update (assuming 0 delay between bytes, which is not likely due to microcontroller processing times). This means that if the SR sends a total of 20 bytes, there is time for 49 from the SPM. 38400 is the fastest data rate which can be accurately generated by most microcontrollers with 8MHz crystals, and it is a rate which can be directly handled by a PC serial port.

Communication is not possible on the Shark Bus with wakeup asserted on SBH, so the SR shall wait until after the end of the wakeup signal before sending its first transmission.

Data shall be sent in packets on the Shark Bus. Packets shall consist of a start byte, which includes the type of the packet and its length; followed by the data; followed by a checksum. The exception to this is the "Transmit Finished" packet, which shall be a single, fixed value byte.

Data consists of Joystick position and switch states from the SR, LED states, flash codes etc from the SPM, and data from other modules as required.

The SR shall initiate the transfer, by sending its "general information" packet. It follows this with other packets (such as HHP data) as required. Finally it sends a "transmit finished" byte. The SPM responds with its own "general information" packet, followed by other packets as required. It also sends a "transmit finished" byte. If there is a SACU in the system, it shall send its data after the SPM has completed.



Packets shall not be re-sent if they are not received correctly; their contents will be discarded by the receiving module in this case.

If the SPM receives 3 consecutive corrupt packets, it shall initiate an emergency stop process by flagging a TRANSIENT fault. It shall not send packets in response to corrupt received packets. This shall also be flagged as a LIMP fault. Once communication with the SR is re-established, the TRANSIENT fault shall be cleared, but the LIMP fault shall remain asserted.

If 60ms elapse without the SPM receiving a valid general packet from the SR (when not in bus idle mode), the SPM shall initiate an emergency stop process by flagging a TRANSIENT fault. This shall also be flagged as a LIMP fault. Once communication with the SR is re-established, the TRANSIENT fault shall be cleared, but LIMP mode shall remain activated.

While the data format is 8 bits, bit 7 in each byte shall be used to indicate the first byte of a packet. Bit 7 shall be CLEAR (0) for the first byte in a packet, and SET (1) for all other bytes. If there is a slight timing difference between the sending and receiving modules, such that the receiver is slightly faster than the sender, having bit 7 set in the majority of the data bytes will give the greatest chance of correct reception.

This means that each transmitted byte carries 7 bits of actual information.

The packets shall consist of a start character which specifies the type of packet and its length. This shall be followed by the data bytes, and then a single byte checksum. The length shall not include the start character or the checksum. The modules shall start a timer upon recognition of a byte, and if 3ms elapse without receipt of a byte and without the packet being completed, the partial packet shall be discarded.

Similarly, if 3ms elapse after the end of a packet without either the start of a new packet, or a "transmit finished" byte, the receiving module shall assume that the transmission has ended. In this case, any packets that were received correctly shall still be regarded as valid.

The Shark Bus does not provide any mechanism for detecting or handling transmit collisions. If these happen all data will be corrupted.

3. Power-up behaviour

At power-up, the SR shall send its Power-up information packet, followed by a "Transmit Finished" byte.

If the SPM does not receive this packet correctly, it shall not respond. The SR shall continue to send its Power-up information packet at 20ms intervals.

If the SPM receives the SR Power-up information correctly, it shall send its own Power-up information packet, followed by a "Transmit Finished" byte.

When the SR receives the SPM's packet correctly, it knows that its own Power-up information packet has been received correctly. It shall stop sending its power-up information packet, and start the normal transmit handling.
When the SPM receives a General Information packet correctly from the SR, it knows that its Power-up information has been received correctly. It shall stop sending its power-up information packet, and start the normal transmit handling.

After 1 second, if the SR has not received a General Information packet successfully from the SPM, it shall indicate a major communications fault, and continue to send packets every 20ms.

After the power-up sequence has been completed and the SPM has received a General packet from the SR, no more power-up packets shall be sent. If the SPM receives a power-up packet from the SR after power-up has been completed, it shall flag a TRANSIENT fault, enter LIMP mode and enter the power-up sequence again (ie. respond to the SR with a power-up packet).

Note about capabilities:

As of version 5 of the Shark Bus communication specification, modules shall report capabilities in their power-up packets. The specific protocol for this is specified below section 5.

Lighting:
To enable lighting functionality, both the SPM and the SR are required to support lighting.
If the SR supports lighting but the SPM does not, the SR shall disable all of its lighting functionality, ignore all lighting-related status information from the SPM, and never request any lighting functionality.
If the SPM supports lighting but the SR does not, the SPM shall disable all of its lighting functionality, ignore all incoming lighting requests and send the "inactive" state for all lighting-related status outputs.

Actuators:
To enable actuator functionality, both the SPM and the SR are required to support actuators.
If the SR supports actuators but the SPM does not, the SR shall disable all of its actuator functionality, ignore all actuator -related status information from the SPM, and never request any actuator functionality.
If the SPM supports actuators but the SR does not, the SPM shall disable all of its actuator functionality, ignore all incoming actuator requests and send the "inactive" state for all actuator-related status outputs. Specifically, if the SR requests an actuator Operating Mode, the SPM shall respond with "Requested operating mode is invalid" - see SPM General Information packet, below.

The preceding comments apply equally to the SACU.

4. Power-down behaviour

The SR sends the SPM its power button state. When the SPM receives a button state that it interprets as a request to power down, it shall set the "Power Down Now" bit in its General Information packet to the SR. Note that the "power down now" bit shall not be set until the power button is **released.** The SPM may also set the "Power Down Now" bit for other reasons, such as sleep.

In response to the SPM's Power Down Now bit, the SR shall set its "Power Down / Sleep Mode Confirm" bit.

When the SPM sees the SR's "Power Down / Sleep Mode Confirm" bit, it shall physically power down.

If communication is lost with the SR without the "power down now" bit set, the SPM shall power down after 5000ms.

5. Packet Structures

The first, or start, byte of a packet defines its type and the length of the data that it contains. The type is in bits 3-0, and the length is in bits 6-4. This allows for 16 packet types, each containing up to 8 bytes of data (the length field does not include the start byte nor the checksum). Note that the number in the length field is one less than the data length: 000 = 1 data byte, 111 = 8 data bytes. The "Transmit Finished" byte is an exception to this, as it has a length field of 000 but has zero data bytes and no checksum.

The data length for a "Transmit finished" byte is 0. It is the first and only byte of the Transmit Finished packet, so bit 7 is clear. The Transmit Finished byte has a fixed value of 0x0F.

The data follows the start byte, in a format defined below.
Finally all packets except for the Transmit Finished packet shall include a one-byte checksum, which is defined as 0x7F - (least-significant 7 bits of (sum of all data bytes and start byte)).

The reason for having a Transmit Finished packet is to minimise latency - each module knows when the preceding module has no more packets to send.

If a module receives a packet of a type that it does not recognise, it shall discard that packet.
If a module receives a packet type that it recognises but the length is greater than it expects, it shall accept that packet and discard the extra information (as long as the checksum is correct).
If a module receives a packet type that it recognises but the length is less than it expects, it shall reject that packet (treat it as corrupt) even if the checksum is correct.

Unused bits in the packets shall be set to zero. In general, all bits are set when active and reset when inactive.

Bus utilisation with packets as described below is as follows:

Immediately after power-up with SACU in system

10 bytes	SR power-up information
1 byte	SR "transmit finished" byte
3 bytes	SPM power-up information
1 byte	SPM "transmit finished" byte
9 bytes	SACU power-up information
1 byte	SACU "transmit finished" byte

Total: 25 bytes

Assuming 50us latency between transmission of each byte, this is 8.41ms or 42.1% utilisation.

Normal operation, no HHP traffic

9 bytes	SR General Information
1 byte	SR "transmit finished" byte
8 bytes	SPM General Information
1 byte	SPM "transmit finished" byte

Total: 19 bytes

Assuming 50us latency between transmission of each byte, this is 6.39ms or 32.0% utilisation.

Normal operation with maximum HHP traffic, joystick calibration, and SACU in system

9 bytes	SR General Information
10 bytes	SR HHP data
3 bytes	SR Joystick calibration
1 byte	SR "transmit finished" byte
8 bytes	SPM General Information
10 bytes	SPM HHP data
1 byte	SPM "transmit finished" byte
7 bytes	SACU General Information
1 byte	SACU "transmit finished" byte

Total: 50 bytes

Assuming 50us latency between transmission of each byte, this is 16.82ms or 84.1% utilisation.

The message types are:

Type	Description
00	SR General Information
01	SPM General Information
02	SR HHP data
03	SPM HHP data
04	SR power-up information
05	SPM power-up information
06	SR Joystick Calibration information
07	SR Factory Test
08	SACU General Information
09	SACU power-up information
0A	SPM Programmable Settings
0B	
0C	

0D	
0E	
0F	"Transmit Finished"

Further descriptions of message contents:

Type 00 **SR General Information**

This packet contains the joystick speed and direction (10 bit readings), speed pot setting (8 bit reading), and input and status bits.

The minimum allowable length for this packet is 6 data bytes.

If the SPM receives only bytes 0-5, and the checksum is correct, the SPM shall assume that it is operating with a previous version of the Shark Remote and take the following actions:

- Disregard the Operating Mode in byte 5
- Disable all lighting and actuator functionality.

Note that these actions shall be taken regardless of whether the SR indicated in its power-up packet that it has lighting and/or actuator functionality.

Byte 0: Joystick speed reading (7 MSbs)

Byte 1: Joystick direction reading (7 MSbs)

Byte 2: Speed pot reading (7 MSbs)

Byte 3:

bit 6: Speed pot reading LSb

bits 5-3: Joystick speed reading (3 LSbs)

bits 2-0: Joystick direction reading (3 LSbs)

Byte 4:

bit 6: Joystick Error (indicates joystick mirror fault or some such problem when set).

bit 5: Speed pot Error (indicates out-of range error or some such problem when set).

bit 4: Local fault (such as CPU fault) (set when there is a fault)

bit 3: Battery charger inhibit state (set when inhibit is active)

bit 2: Power switch state (all bits in this byte are 1 for active, 0 for inactive).

bit 1: Horn switch state (set when switch is pressed)

bit 0: Lock switch state (set when switch is pressed)

Byte 5:

bit 6: Hazard light request

bit 5: Left Indicator request

bit 4: Right Indicator request

bit 3: Remote Inhibit. When this bit is 1, the SPM shall not drive.

bit 2: Programmer Comms flow control. Set when SPM MAY send HHP packets, clear when buffer space is low.

bit 1: Joystick Calibration active (set when JC mode is active)

bit 0 : Power down / Sleep mode confirmation. Set when Sleep or Power down has been requested and the SR is ready to comply.

Byte 6:

bit 6: Not used

bit 5: Local non-critical fault (causes "limp" mode)

bit 4: Headlight request

bits 3-0: Operating mode, defined as

00	Drive mode
01	Actuator 1 mode
02	Actuator 2 mode
03	Actuator 1+2 mode
04-0F	currently undefined

Notes:

The battery charger inhibit state is a filtered value which shows "active" continuously when data is being transmitted to the HHP. The SPM may disregard this and allow driving when the HHP is active, to allow driving while programming.

When the SR indicates a joystick error, it shall also send neutral joystick values.

When the SR indicates a Speed Pot error, it shall also send a mid-range speed pot value.

When Joystick Calibration mode is active, the SR shall also send packet type 06: "Joystick calibration".

When the SR wishes to select an operating mode other than the currently selected one, it transmits the appropriate value in bits 3-0 of byte 6. The SPM responds with the same value in its Operating Mode value, but may also set its "Requested operating mode is invalid" flag. If this occurs, the SR must change its requested operating mode. This document does not specify which operating mode the SR must change its request to; it may be another operating mode or it may be 00 (Drive Mode).

Drive mode is always valid. The SPM shall never set its "Requested operating mode is invalid" flag while transmitting 00 as its operating mode; however if the SR receives this due to some error, it shall disregard the "Requested operating mode is invalid" flag.

Once the SPM has transmitted the requested operating mode without the "Requested operating mode is invalid" flag, and the SR has received this, the operating mode is established. The SPM shall not subsequently set the "Requested operating mode is invalid" flag until the operating mode is changed.

Type 01 SPM General Information

The minimum allowable length for this packet is 7 data bytes.

Byte 0:

bit 6: Lock Status (set for locked, reset for unlocked)

bit 5: DCI Slowdown (set for slow, reset for normal)

bits 4-0: Fuel gauge value (0 = empty, 18 = full)

Byte 1:

bit 6: Drive Inhibited (set for inhibited, reset for drivable)

bit 5: Display disable (set for disable, reset for enable)

bit 4: Power down now (set when power-down requested)

bits 3-0: System state, defined as (from the Requirements Specification)

00:	System OK
01:	User fault
02:	Battery fault - "replace batteries"
03:	Left motor fault
04:	Right motor fault
05:	Left Park Brake fault
06:	Right Park Brake fault
07:	Shark Remote fault
08:	Shark Power Module fault
09:	Shark cable fault
0A:	Unknown fault

Byte 2:

bit 6: Lock mode enable - set if lock mode is enabled, clear if lock mode disabled

bit 5: SR Temporary drive modification - set for OONAPU test and Stall timeout, reset otherwise.

bit 4: Joystick wakeup from sleep allowed (set when joystick wakeup is allowed)

bit 3: Sleep mode requested (set when active)

bit 2: Joystick calibration mode (set for calibration, reset for normal operation)

bit 1: not used

bit 0: Sound horn (set when horn required)

Byte 3:

bit 6: Attendant / User indication: 1 = SACU is driving control, 0 = SR is driving control

bit 5: System is driving (this includes "driving an actuator").

bit 4: Lighting flash cadence - toggles in time with the actual lighting outputs from the SPM.

bit 3: Hazard light status (1 = light active, 0 = light off)

bit 2: Left Indicator status

bit 1: Right Indicator status

bit 0: Headlight status

Byte 4:

bit 6: SACU joystick calibration mode

bit 5: SACU Temporary drive modification - set for OONAPU test and Stall timer, reset otherwise

bit 4: SACU power-up acknowledgement

bit 3: Hazard light fault

bit 2: Left Indicator fault

bit 1: Right Indicator fault

bit 0: Headlight fault

Byte 5:

bit 6: Actuator 2 fault

bit 5: Actuator 1 fault

bit 4: SR requested operating mode is invalid

bits 3-0: Operating Mode, defined as

00	Drive mode
01	Actuator 1 mode
02	Actuator 2 mode
03	Actuator 1+2 mode
04-0F	currently undefined

Byte 6:

bit 6: Unused

bit 5: SACU requested operating mode is invalid

bits 4-0: Speedo reading

This is the SPM's best estimate of the actual ground-speed of the wheelchair. Its value is 0 when the chair is stopped, and 31 when the chair speed is 100% of the currently active maximum.

Notes:
When the SR or SACU sees its "Joystick Calibration Mode" set, it shall carry out its internal joystick calibration procedure. It shall continue to send joystick values in its general information packets during this procedure (these may vary due to the calibration process). The SPM shall inhibit driving during joystick calibration, and set an OONAPU state after calibration.

When the SR wishes to select an operating mode other than the currently selected one, it transmits the appropriate value in bits 3-0 of byte 6. The SPM responds with the same value in its Operating Mode value, but may also set its "Requested operating mode is invalid" flag. If this occurs, the SR must change its requested operating mode. This document does not specify which operating mode the SR must change its request to; it may be another operating mode or it may be 00 (Drive Mode).
Drive mode is always valid. The SPM shall never set its "Requested operating mode is invalid" flag while transmitting 00 as its operating mode; however if the SR receives this due to some error, it shall disregard the "Requested operating mode is invalid" flag.
Once the SPM has transmitted the requested operating mode without the "Requested operating mode is invalid" flag, and the SR has received this, the operating mode is established. The SPM shall not subsequently set the "Requested operating mode is invalid" flag until the operating mode is changed.

Type 02 SR HHP data
Type 03 SPM HHP data

"HHP" is a generic term, used to refer to the programming device that is communicating with the system through the SR Battery Charger port. It specifically includes the DX-HHP, the Mark 4 HHP, and the Dynamic Wizard; as well as any suitable equipment.
These packets carry data from the SPM to the HHP. The SR is a transparent intermediary. It shall receive and buffer data from the HHP and send it (when able) to the SPM in packet type 02. When it receives packet type 03, it shall immediately transmit the data bytes to the HHP.
These packets may carry up to 8 bytes of data. This allows for a complete HHP screen redraw (100 characters) in less than 0.5 seconds.

The Invacare HHP uses several characters with codes greater than 0x7F. These cannot be transmitted directly over the Shark Bus, so the following translation shall be performed by the SR:

1) For characters transmitted from the HHP to the SPM
Characters over 0x80 shall have 0x70 subtracted. These characters are always 0x80 to 0x83, so the resulting codes are 0x10 to 0x13.

2) For characters transmitted from the SPM to the HHP:
Characters with codes 0x10 to 0x17 shall have 0x70 added, making their codes 0x80 to 0x87.
Characters with codes 0x18 to 0x1F shall have 0xE0 added, making their codes 0xF8 to 0xFF.

Type 04 SR power-up information

The minimum allowable length for this packet is 6 data bytes. If the SPM receives this packet with a length less than 8 data bytes, it shall assume that the SR does not support actuator or lighting control.

Byte 0: Remote type: Types recorded elsewhere (see 📄)
Byte 1: Year of manufacture, minus 2000 - for example, 01 is 2001, 127 is 2127.
Byte 2: bits 3-0: Month of manufacture (1 is January, 12 is December).
Byte 3: Serial number bits 20-14
Byte 4: Serial number bits 13-7
Byte 5: Serial number bits 6-0
Byte 6: Software version number
Byte 7: Capabilities:
bits 6-3: currently unused
bit 2: If 0, has a virtual speed pot. If 1, has an analog speed pot. This flag is only honoured where the software version is 2.5 or greater (whether prerelease or not)
bit 1: Supports 2 actuator control
bit 0: Supports lighting control

Notes:
The serial number format is chosen for compatibility with Dynamic Controls' standard serial number format (YYMXXXXX).
Allowance has been made such that the maximum serial number for any month can go up to 1,048,576. (they start at 10,000)

The values for the Software version number are:
0x00 Versions before 1.0 release. This version has a 7-byte power-up packet, it omits the build version number.
0x01 Release version 1.0. This version has a 7-byte power-up packet, it omits the build version number.
0x02 Version 1.1 up to release candidate 1. No released product uses this value.

Where the upper nibble of the version number is greater than zero, the format of the version number is:
Bits 6-4: Software major version
Bit 3: "Pre-release flag": when set, this software is pre-release and is not suitable for production.
Bits 2-0: Software minor version.
For example, software version 1.1release candidate 2 will have a software version number of 0011001b or 0x19. Version 3.2 release will have a software version number of 0110010 or 0x32.
The format is chosen to enable easy reading of a hex dump such as that produced by the ^ terminal command on the Shark Power Module (see SPM SRS section 26.14 for details).

The maximum version number that can be indicated using this scheme is 7.7.

Type 05 SPM power-up information

Byte 0: Capabilities:
bits 6-3: currently unused
bit 2: Supports Actuator 2 control
bit 1: Supports Actuator 1 control
bit 0: Supports lighting control

Type 06 Joystick Calibration (SR or SACU)

Byte 0:
bit 6: Not used.
bit 5: Second pass of joystick calibration
bit 4: Neutral seen at end of calibration process.
bit 3: Forward deflection seen during calibration process.
bit 2: Reverse deflection seen during calibration process.
bit 1: Left deflection seen during calibration process.
bit 0: Right deflection seen during calibration process.

Type 07 Factory Test

This packet type is purely for use in production testing (and design testing). The protocol for using it is a little different from the other packets, because IT IS INTENDED TO BE USED WHEN THERE IS NO SPM IN THE SYSTEM. Another device (such as a PC or production test fixture) sends a request packet to the SR or SACU, which shall respond once only with the information requested. If the SPM ever receives a Type 07 packet, it should log an information-level event and ignore the packet.
Note that the SR/SACU does not respond to the Reset Micro command, it simply resets immediately.

Packet 07 sent TO the SR or SACU

Byte 0: bit 6: MSb of Address
bit 5: MSb of Data
bit 4: Module Select - 0 for SR, 1 for SACU
bits 3-0: Command
Byte 1: Address bits 6-0
Byte 2: Data bits 6-0

The commands are:
00 Read register space (addresses 0x00 to 0xFF valid)
01 Read static RAM space (addresses 0x00 to 0xFF valid)
02 Read EEPROM (addresses 0x00 to 0xFF valid)
03 Write register space
04 Write static RAM space
05 Write EEPROM
06 Reset micro
07 Set SRAM Page to the value in the Data byte. This page value shall be used for all 01 and 04 commands until a further 07 command is received. Allowable values are 0x00 to 0x03 for the ATmega8. If a value greater than the maximum allowable is specified, the command shall fail: the active SRAM page shall not be changed and the data byte shall be changed to 0xFF in the response.
08 Select device. This is used to select which micro to send the factory commands to when the remote contains more than one. A value of 0 will select the micro servicing the SharkBus. It is upto this micro how and if it relays the factory commands to other micros within the remote, while they are selected. It is also responsible for relaying replies to factory commands from other micros back to the SPM.

For READ operations, the Data byte is don't care. For RESET, the Address and Data bytes are don't care. For the SELECT DEVICE operation the Address byte is ignored.
Note: the SR shall not respond to a factory packet where the Module Select flag is set. The SACU shall not respond to a factory packet where the Module Select flag is clear.

Packet 07 sent BY the SR

Byte 0: Command (same as received, but with data MSb changed for read operations)
Byte 1: Address (same as received)
Byte 2: Data (for Write operations, same as received. For Read operations, the data that was requested)

A note about addressing:
For accesses to Register and SRAM space, the AVR CPU's data indirect addressing mode is used: the address specified is loaded into the Y register, and an LD or ST instruction is performed as appropriate. The address handling is different for the Register and SRAM space:

- For the Register space command, the address is used unchanged. This means that addresses 0x00 to 0x1F access the

AVR CPU's general purpose registers, 0x20 to 0x5F access the AVR CPU's special function registers, and 0x60 to 0xFF address Static RAM space.

- For the SRAM space command, the effective address combines the address supplied with the command (0x00 to 0xFF) with the SRAM Page setting (0x00 to 0x03) as the high byte, to produce 0x000 to 0x3FF. This value then has 0x60 added to it, to produce a result of 0x060 to 0x45F, which is the address used for the LD or ST instruction.

Type 08 SACU General Information

This packet has similar intent to the SR General Information packet, conveying joystick and speed pot positions. If the SACU does not implement speed control, it shall set its speed pot value to zero.

Byte 0: Joystick speed reading (7 MSbs)
Byte 1: Joystick direction reading (7 MSbs)
Byte 2: Speed pot reading (7 MSbs)
Byte 3:
bit 6: Speed pot reading LSb
bits 5-3: Joystick speed reading (3 LSbs)
bits 2-0: Joystick direction reading (3 LSbs)
Byte 4:
bit 6: Joystick Error (indicates joystick mirror fault or some such problem when set).
bit 5: Speed pot Error (indicates out-of range error or some such problem when set).
bit 4: Local fault (such as CPU fault) (set when there is a fault)
bit 3: Joystick Calibration active (set when JC mode is active)
bit 2: SACU User/Attendant switch: 1 = Attendant, 0 = User.
bit 1: Local non-critical fault (causes "limp" mode)
bit 0: SACU Drive Inhibit
Byte 5:
bits 6-4: Not used
bits 3-0: Operating mode, defined as

00	Drive mode
01	Actuator 1 mode
02	Actuator 2 mode
03	Actuator 1+2 mode
04-0F	currently undefined

Notes:
When the SACU indicates a joystick error, it shall also send neutral joystick values.
When the SACU indicates a Speed Pot error, it shall also send a mid-range speed pot value.
When Joystick Calibration mode is active, the SACU will also send packet type 06: "Joystick calibration".

See "SR General Information" packet, above, for details on the protocol for changing Operating Modes.

Type 09 SACU power-up information

Byte 0: Year of manufacture, minus 2000 - for example, 01 is 2001, 127 is 2127.
Byte 1: bits 3-0: Month of manufacture (1 is January, 12 is December).
Byte 2: Serial number bits 20-14
Byte 3: Serial number bits 13-7
Byte 4: Serial number bits 6-0
Byte 5: Software version number
Byte 6: Capabilities:
bits 6-3: currently unused
bit 2: Supports Actuator 2 control
bit 1: Supports Actuator 1 control
bit 0: Supports speed control

The serial number format is chosen for compatibility with Dynamic Controls' standard serial number format (YYMXXXXX). Allowance has been made such that the maximum serial number for any month can go up to 1,048,576. (they start at 10,000)

The encoding of the Software version number is as follows:
Bits 6-4: Software major version
Bit 3: "Pre-release flag": when set, this software is pre-release and is not suitable for production.
Bits 2-0: Software minor version.
For example, software version 1.1release candidate 2 will have a software version number of 0011001b or 0x19. Version 3.2 release will have a software version number of 0110010 or 0x32.

Type 0A SPM Programmable Settings

This packet shall be transmitted by the SPM at least once per second. It contains user-adjustable data required by the other modules.

Byte 0:

bit 6: Not used.
bit 5: Speedo Option: 0 is "Max-speed only", 1 is "Speedo + Max-speed"
bit 4: "Click on keypress" enabled
bits 3-0: Speed button sensitivity, legal values are 1 - 10.

Byte 1:

bits 6-0: Joystick switch threshold, legal values are 1 - 100.

<u>Type 0F</u> <u>Transmit Finished</u>

Once printed this document is uncontrolled, once attachments are placed outside this document, these files are uncontrolled.
It is the responsibility of the user of this printed document or its attachments to ensure that the content is current and valid !