NSP Packet Protocol for Sinclair Interplanetary Hardware
Rev 1.0
February 11 2008

1. Scope

This document is intended to be read by users of Sinclair Interplanetary satellite components that use the common NSP communications protocol. It describes how NSP is used inside a spacecraft to move data to and from sensors and actuators.

The full and definitive NSP document is available from Daniel Kekez at UTIAS/SFL. It contains additional features (such as space-Earth links) not used by Sinclair Interplanetary equipment and thus not of interest to the target reader.

2. Concept

NSP is the Nanosatellite Protocol, originally developed at UTIAS/SFL for use on the CanX nanosatellites. This in turn is descended from the Simple Serial Protocol (SSP) used by UTIAS/SFL and Dynacon on the MOST and CHIPSAT spacecraft as well as the Dynacon reaction wheels in the wider market.

These protocols were developed as more and more microprocessors found their way into small spacecraft. Each satellite component will typically contain a processor, and it is convenient to manage all of its command and telemetry requirements using digital communications. In contrast to a more traditional set of analog telemetry and high-level command lines, digital links require fewer wires in the harness and are less susceptible to noise. NSP specifies a digital link standard that is used by many pieces of Sinclair Interplanetary equipment. Use of the same standard on multiple devices allows reuse of code on the part of the customer and significant savings in GSE.

The basic unit of communication in NSP is the message. A message from the satellite on-board computer (OBC) to a Sinclair Interplanetary device carries a single telecommand. The device may then reply to the OBC with a single response message. This may be a simple acknowledgement, or it may contain additional telemetry.

NSP messages can be carried on a number of different types of link: asynchronous, synchronous and I2C. Large NSP networks may contain different types of links with routers and bridges to connect them.

3. Message Format

3.1. Byte Stream

NSP messages are composed of an ordered sequence of 8-bit bytes. Wherever multiple adjacent bytes are grouped together to form larger storage units (16-bit integers, 32-bit floats, etc) the least significant bytes are transmitted and received first.

3.2. Message Fields

Length	Field
1 byte	Destination Address
1 byte	Source Address
1 byte	Message Control Field
0 or more bytes	Data
2 bytes	Message CRC

The table above shows the fields that make up an NSP message. The data field has variable length. If its length is zero (the message carries no data) then the total message contains five bytes. The maximum number of data bytes that can be placed in a message depends on the NSP implementation in each particular device. 260 bytes is a common limit.

3.3. Message Control Field

Bit 7 (MSB)	"Poll" Bit
Bit 6	"B" Bit
Bit 5	"A" (ACK) BIT
Bits 4 – 0	Command code

Within each NSP message is a message control field byte. The table above shows how the individual bits in this byte are decoded. There are three Boolean quantities, labeled "A", "B" and "Poll". There is also a five bit command code, interpreted as an unsigned integer between 0 and 31.

3.4. Message CRC

Each NSP message contains a 2 byte (16-bit) CRC to guard against errors in transmission. The 16-bit CCITT polynomial is used: x^16 + x^12 + x^5 + 1. The initial shift register value is 0xFFFF. Bytes are fed into the CRC computation starting with the destination address, and concluding with the last byte of the data field. Within a byte, bits are fed in LSB first.

The following fragment of C code, courtesy of Henry Spencer, illustrates how the CRC can be computed. It is worth noting that most Sinclair Interplanetary devices actually perform an identical calculation but using a hardware CRC engine.

```
#define POLY 0x8408 /* bits reversed for LSB-first */
unsigned short crc = 0xffff;
while (len-- > 0) {
        unsigned char ch = *bufp++;
        for (i = 0; i < 8; i++) {
                crc = (crc >> 1) ^ ( ((ch ^ crc) & 0x01) ? POLY : 0 );
                ch >>= 1;
        }
}
```

4. Telecommand Format

The OBC will send telecommands to Sinclair Interplanetary devices. These messages have the following attributes:

Destination Address	The address of the SI device in question
Source Address	The unique address of the OBC
"Poll" bit	'1' If a reply is requested '0' If no reply is desired
"B" bit	Ignored by SI device (But see Reply Format)
"A" bit	Ignored by SI device
Command Code	Desired command
Data	Data bytes, as appropriate for command
CRC	CRC, computed from the rest of the message

The list of command codes, and the formatting of the data to accompany them, is specific to the individual Sinclair Interplanetary device.

5. Telecommand Validation

When a sequence of bytes that may be a telecommand are received, a Sinclair Interplanetary device will take the following steps to validate it:

- The destination address will be compared to the set of addresses that the device will accept. Details of addressing are covered in a later section.
- The message must meet the minimum message length (5 bytes)
- The message must not exceed the maximum message length, determined by the device implementation.
- There must not have been an error in SLIP framing (see SLIP section for details).
- The message CRC must be good.

Messages that fail any of these criteria are rejected and forgotten. Valid telecommands are passed on to the next software layer for execution.

6. Reply Format

After a telecommand has been executed, a Sinclair Interplanetary device will generate a reply message if (and only if) the telecommand "Poll" bit is '1'. The reply will have the following attributes:

Destination Address	The source address from the telecommand
Source Address	The destination address from the telecommand
"Poll" bit	'1' in all cases
"B" bit	The "B" bit from the telecommand
"A" bit	'1' if the device reports an ACK condition '0' if the device reports a NACK condition
Command Code	The command code from the telecommand
Data	The data bytes from the telecommand, followed by zero or

	more bytes of telemetry
CRC	CRC, computed from the rest of the message

It can be seen that much of the reply message is a direct echo of the telecommand message. This makes inefficient use of the link bandwidth, but makes the context of each reply completely unambiguous.

7. SLIP Framing

NSP messages are encapsulated into packets using SLIP framing before being transmitted to another device. Upon reception the framing is removed. SLIP framing is described in RFC 1055.

A special character FEND (0xc0) marks both the beginning and end of each NSP message. Wherever FEND would occur within the message, it is replaced by two bytes: FESC TFEND (0xdb 0xdc). Wherever FESC would occur within the message, it is replaced by FESC TFESC (0xdb 0xdd).

When processing a SLIP framed message, it is an error to see FESC followed by anything other than TFESC or TFEND.

SLIP framing is required because not all of the link types supported by Sinclair Interplanetary feature out-of-band message completion signals.

8. NSP Addresses

NSP addresses are eight bits long. Recommended address ranges are between 1 and 127. Any user of Sinclair Interplanetary devices should pick an NSP address for the spacecraft OBC (or GSE test computer). By convention, 0x11 is used as the NSP address for the primary computer.

Some Sinclair Interplanetary devices have unique NSP addresses programmed into them in the factory. Others have pins on their connectors that can be shorted to ground to select between a number of different addresses. Finally, devices operating on synchronous (SPI) busses use out-of-band addressing and do not have a fixed NSP address. Consult the particular documentation for your device to determine its NSP address capabilities.

9. Asynchronous Encapsulation

Certain Sinclair Interplanetary devices are configured for asynchronous communications. They have one transmit and one receive channel. This may operate at +3.3 V or +5 V logic levels, or it may use an RS-485 driver.

Both channels operate at the same baud rate. Consult your device documentation for its baud rate and tolerance. There are 8 bits per byte, no parity, and 1 stop bit (8N1).

The OBC sends telecommands into the receive channels of devices. Since each device has a unique NSP address, all of the receive channels can be wired together if desired. Telecommand bytes can be sent as rapidly as desired (limited only by the baud rate) or as slowly as desired. There is no requirement that a message be completed in a particular period of time.

A reply message (if "Poll" bit is '1') will occur quickly following the reception of the telecommand. The device will send out the reply on its transmit channel. At the conclusion of the reply (after the stop bit has been transmitted on the final FEND character) the device will turn off its output channel driver leaving only a weak pull-up. This permits multiple devices to have their transmit channels directly wired together.

A device can process only one telecommand at a time. Once a telecommand has been received, the link's receive channel is disabled and incoming bytes are ignored. The receiver is re-enabled only when the telecommand execution is complete, and the reply (if any) has been transmitted.

Both the telecommand message and the reply message (if any) are framed using SLIP.

10. Synchronous Encapsulation (SPI slave)

[Basic familiarity with the SPI standard is required to understand this section. See http://en.wikipedia.org/wiki/Spi for background.]

Some Sinclair Interplanetary devices are configured for synchronous communications (SPI slave). Each device has a Master-In-Slave-Out (MISO) pin, a Master-Out-Slave-In (MOSI) pin, a Slave Select (/SS) pin and a Serial Clock (SCK) pin. The MISO, MOSI and SCK signals are common for all of the devices on the SPI bus. However, each device has its own /SS signal, connected directly to the OBC.

MOSI, SCK and /SS are always inputs on the Sinclair Interplanetary device. When /SS is at a logic low, MISO is an output. Otherwise it is high impedance with a weak pull-up. The device outputs new data on MISO on the falling edge of SCK, and samples data from MOSI on the rising edge of SCK.

A device receives and transmits data when /SS is low. /SS must be low for the entire duration of an 8-bit byte transfer. The OBC may drive /SS high between bytes, or /SS may remain low for the entire duration of a telecommand/response transaction. /SS should not be wired low, even in applications where there is only one device on the SPI bus, as noise on SCK can then cause the device to become irrecoverably confused about where bytes begin and end.

Messages are SLIP framed as usual. They begin and end with FEND, and instances of FEND and FESC inside them are escaped. The SLIP framing protocol is also expanded to compensate for the full-duplex nature of the bus.

On an SPI connection one bit is transferred from device to OBC, and one bit from OBC to device, each SCK cycle. Thus, while receiving a telecommand the device is required to transmit something. The SLIP framing character FEND (0xc0) is sent as each telecommand byte is received.

The SCK signal is controlled by the OBC, so the Sinclair Interplanetary has no control of when it transmits and receives bytes. Consequently the OBC must poll the device when it expects a reply packet. The OBC does this by sending repeated FEND characters following the end of its telecommand message. The device will respond with FEND as it executes the telecommand, and will then begin to respond with sequential bytes from the reply packet. Once the reply packet is finished the device will once again respond with FEND characters until the next telecommand is received.

If the device receives bytes while busy with telecommand execution it will ignore them – it can only process one telecommand at a time. However, if a byte that is not FEND is received while the device is transmitting its reply message the reply is immediately aborted and the device begins to receive the new telecommand.

Devices on an SPI bus are hardware addressed by their /SS pins. This takes priority over the destination address field in telecommands. Bytes are not received when /SS is high, and all bytes received when /SS is low are considered part of a telecommand addressed to the receiving device. For ease of software compatibility, NSP messages still contain source and destination address fields. The Sinclair Interplanetary device will simply swap the two addresses in its reply message but will otherwise ignore them.

11. I2C Encapsulation

[Basic familiarity with the I2C standard is required to understand this section. See http://en.wikipedia.org/wiki/I2c for basic background.]

The final link option available for Sinclair Interplanetary NSP devices is I2C. The devices are never bus-masters, and all transactions are initiated by the OBC. The NSP address of a device is also used as its I2C address – note that I2C addresses are only 7 bits long, while NSP implemented on other links will accept up to 8 bits.

NSP telecommand over I2C with no reply		
Transmitter	Data	Notes
OBC	START	
OBC	Slave address, Write	Slave ACK indicates slave receiving
OBC	Source address	
OBC	Message Control Field	Poll bit = 0
OBC	Outgoing data (0 – N bytes)	
OBC	16-bit CRC	
OBC	FEND	Slave ACK only if CRC valid
OBC	STOP	

The table above shows the OBC sending a telecommand to a device where no reply is required. The message is SLIP framed, except that the leading FEND is omitted as it is redundant – I2C provides an out-of-band frame start signal.

The destination address is not transmitted. Instead, the transaction begins with the slave address and the write bit, in accordance with the I2C specification. Note that the CRC is computed based on the NSP message, not on the bytes transmitted over the I2C link! When the final FEND is received the device calculates the CRC for the packet and sets the I2C ACK bit only if the CRC is valid.

NSP telecommand over I2C with reply		
Transmitter	Data	Notes
OBC	START	
OBC	Slave address, Write	Slave ACK indicates slave receiving

OBC	Source address	
OBC	Message Control Field	Poll bit = 1
OBC	Outgoing data (0 – N bytes)	
OBC	16-bit CRC	
OBC	FEND	Slave ACK only if CRC valid
OBC	START	
OBC	Slave address, Read	Slave ACK if reply packet available
Device	Message Control Field	Poll bit = 1
Device	Reply data (0 – N bytes)	
Device	16-bit CRC	
Device	FEND	
OBC	STOP	

The table above shows the case of the OBC sending a telecommand to a device where a reply is required. Instead of sending a STOP at the end of the telecommand, the OBC sends a repeated START. To comply with the I2C specification it must then re-send the slave address in read mode.

The slave then sends the reply message. This is SLIP framed, but as before the leading FEND is omitted. The destination address and source address are also both omitted. An I2C transaction is atomic, and so there can be no question of which telecommand produced which reply. Even though these fields are omitted, the CRC is still computed based on the entire NSP message in the normal fashion.

Sinclair Interplanetary devices use SCL clock stretching for flow control. While SCL is normally driven by the bus master, the device may hold it low while its internal software decides what to do next. The longest such bus stall will be just after the second slave address is received when a reply is requested. The internal processor must execute the incoming telecommand and format the reply before it releases SCL.

Sinclair Interplanetary devices implement the SMBus SCL timeout. If SCL is held low for longer than 25 msec the device will reset its I2C state.