Cygnus Manual

Cygnus Manual

Revision History

Style Sheet is: X:\Company\SmartMasters\Orion

Filenames are:
X:\Company\Manuals\ReferenceManual\Components\Cygnus\RevA\CygnusManual.lwp
X:\Company\Manuals\ReferenceManual\Components\Cygnus\RevA\CygnusTitlePage

Rev	Date	Author	Description
A	16 Aug 99	Emil Farkas	Initial release.

Approvals: Author ________________________________

VP. Engineering ________________________________

Table of Contents

This page intentionally left blank.

1. Introduction

Congratulations on your choice of the Cygnus satellite transceiver. As you use your new Cygnus instrument we know you will appreciate the many features that provide excellent performance.

It is very important to understand how Cygnus operates before using it. On the following pages you will find a wealth of information regarding all aspects of the transceiver. Please read the instructions carefully before installing and operating the equipment. Refer to these instructions every time you want to service a Cygnus unit.

If you have problems or need technical support, please submit requests for technical support by e-mail or fax. This permits you to fully explain your problem and include "evidence" which allows us to submit your problem to the most knowledgeable person for reply and provide you with efficient support.

by e-mail:	**support@nanometrics.ca**
by fax:	**To: Support at fax (613) 592-5929**
by phone:	**Please ask for Support at (613) 592-6776**

Nanometrics Inc.
250 Herzberg Road
Kanata, Ontario Canada
K2K 2A1

This page intentionally left blank.

2. Organization of this Manual

This manual is organized in seven major sections:

Chapter 1	**Introduction**
Chapter 2	**Organization of this Manual**
Chapter 3	**Unpacking and Post Delivery Inspection** Identification of the components you have purchased. It also references an "as-shipped" section.
Chapter 4	**Technical Description** Description of features and technical specifications of the Cygnus.
Chapter 5	**Post-delivery Bench Testing and Configuration** Recommendations foe bench testing the unit in laboratory conditions after being received and before installation.
Chapter 6	**Field Installation** Notes on the field installation of the unit.
Chapter 7	**Servicing** Recommended maintenance and repair procedures, including firmware update instructions.
Appendices	These list mostly tabular material such as error messages, and pin connections.

This page intentionally left blank.

3. Unpacking & Post Delivery Inspection

Introduction

The purpose of the post delivery inspection and tests is to detect any damage which may have occurred during shipment and confirm the basic functionality of the equipment. This test should be performed immediately after receiving each Carina. Further tests are performed prior to field installation and are more detailed, including all of the remote site hardware components. These tests are described in the Remote Site Installation Manual.

Equipment needed

To get started you will need the equipment shown in the table below:

Quantity	Description
1	Power Supply, 10V-15V, 3A or 12V battery
1	IBM compatible personal computer (PC)
1	Carina power cable
1	Data source to Carina cable
1	The Nanometrics serial receiver program (Cygnet)
1	The Libra User Interface program (LUI)
1	Data source which intended to be used with the satellite transceiver.
1	Two twisted pair LAN cables and a LAN hub OR a null LAN cable
1	Spectrum analyser with necessary RF cables, DC blocks and attenuators
1	Trimble GPS antenna mounted in a place with good sky visibility. GPS antenna cable.

Table 1: Equipment required to get started

This startup procedure verifies that the satellite transceiver is running, receives serial data on the serial port and outputs the receive data on the Ethernet port.

Preparation

Install the Libra User Interface program and the serial receiver program (Cygnet) on the computer. For installation instructions refer to the Software Installation Manual. Note, that the Cygnet program is part of the Serial Client package.

Connect the Carina to the PC via an isolated local area network. Connect the Trimble GPS antenna to Carina with the GPS antenna cable. Connect the data source to Carina. Connect the power connector of Carina to a 12V DC power supply and power up the unit.

Inspection

Once the satellite transceiver has finished its self test it will start transmitting data.

Open and inspect the shipment for possible damage. Carefully check each item for damage or defects. The following list includes items generally included with a Carina unit. This list might vary from application to application. To find out the exact list of items included in your shipment refer to the shipping documents (packing list and as-shipped sheet).

The system should have the following contents:

1. Carina satellite transceiver
2. GPS antenna and mounting bracket
3. GPS antenna cable
4. Satellite antenna cables
5. Power cable

6. Satellite antenna
7. Carina manual
8. As-shipped Sheet
9. Release Notes (if applicable)
10. CD or diskette with firmware code (if the transceiver is shipped integrated into a system the firmware is included on the same CD with the system software).

Checking the As-Shipped Sheets

As written, this manual covers the Carina satellite transceiver. Please study the as-shipped data sheet to determine the exact configuration of the instrument. The as-shipped sheet lists the serial numbers of the parts shipped, the exact hardware configuration and calibration parameters associated with your hardware. The as-shipped sheet determines how your Carina unit operates when first turned on. If the satellite transceiver was calibrated to be used with a certain outdoor unit the as-shipped sheet will state the serial numbers of all outdoor hardware with which the given Carina should be used. Several features may have been added to the transceiver since this manual was released. Such new features are described in the Release Notes which have precedence over information in the manual.

Backup

It is strongly recommended that you backup the CD or the diskette and keep a record of all of the post delivery changes made to the hardware and firmware.

Tests

Ping

Start a ping session in an MS-DOS window on the computer with the correct LAN IP address of the actual Carina transceiver. The IP address can be found in the as-shipped sheet of the unit.

Monitor the messages in the ping session window. The transceiver should reply to the packets sent out from the computer.

If the ping is not successful check:

- The connections between the Carina unit and the computer.
- If the IP address of the computer is in the same subnet with the IP address of the Carina LAN interface.
- If you are pinging the right IP address (the IP address of the LAN interface of the Carina unit in the as-shipped sheet).

Contact Nanometrics for support.

Receiving data on the ethernet port

Perform this test only after the Ping test is successful.

Start the Libra User Interface program and in the Operation - Serial ports tab monitor if the number of received good packets is continuously increasing and the number of bad packets is stable. If data is not being received at all check if:

- The data source outputs data.
- The connections between the data source and the Carina are good.

Contact Nanometrics for support.

If the number of frame is continuously increasing check if the serial port is configured for the baud rate at which the data source outputs data.

Transmitting data on the Ethernet port

Start the Cygnet program with the correct command line parameters allowing you to connect to the Carina under test.

Monitor the Cygnet window and check if the received data is correct by comparing it with the output of the data source.

If data is not being received by Cygnet check if you are using the correct IP address and other command line parameters.

Contact Nanometrics for support.

GPS not locking

After startup of the unit in the Operation - GPS tab of the Libra User Interface program monitor the activity on each GPS channel. The signal strength values from the GPS satellites should be greater than 38 dB in order for the GPS to track the satellite. The GPS should search and than track satellites on any of the channels. It needs to track one satellite at least on 5 channels in order for it to provide good time reference to the Phase Lock Loop of Cygnus. Depending on how far are you from where the Cygnus GPS was last time locked (e.g.: Nanometrics offices in Kanata, approx.: 45N/75W) it may take 30 minutes or more for the PLL to be in Fine_Lock state. You can monitor the PLL status in the Operation - GPS Time tab of the Libra User Interface program.

Most problems with the GPS engines can be traced to a poorly placed antenna. If the GPS is not receiving well do the following:

- Reposition the antenna to a location with better visibility of the sky.
- Ensure that the antenna cable is not hanging from the antenna, but is secured with a few tie wraps near the antenna to carry the weight of the cable.
- Check if another GPS with display will lock to satellite. If this GPS receiver does not lock it means that either there is not enough GPS satellite coverage over the area at the time when the test is being performed.

Contact Nanometrics for support.

This page intentionally left blank.

4. Technical Description

Overview of the Hardware

The Cygnus satellite transceiver integrates all the necessary hardware to receive data on two serial ports and transmit the received data to the acquisition center via a satellite link. It includes a high precision GPS timing subsystem, two serial data input ports, an Ethemet port for external access, a satellite modem for both inbound and outbound satellite links and all necessary hardware for the interconnection and operation of all of the above.

The transmission protocol used over the VSAT link is an IP protocol including both the UDP and the TCP as data transfer layers. Essentially each Cygnus Transceiver can be considered as a communications repeater node of a wide area network. The Ethernet port supports several IP protocols such as E-mail, ftp, and telnet.

The following block diagram provides you with a clear view of the different hardware modules integrated in Cygnus and the way it is connected with external components of the remote site.

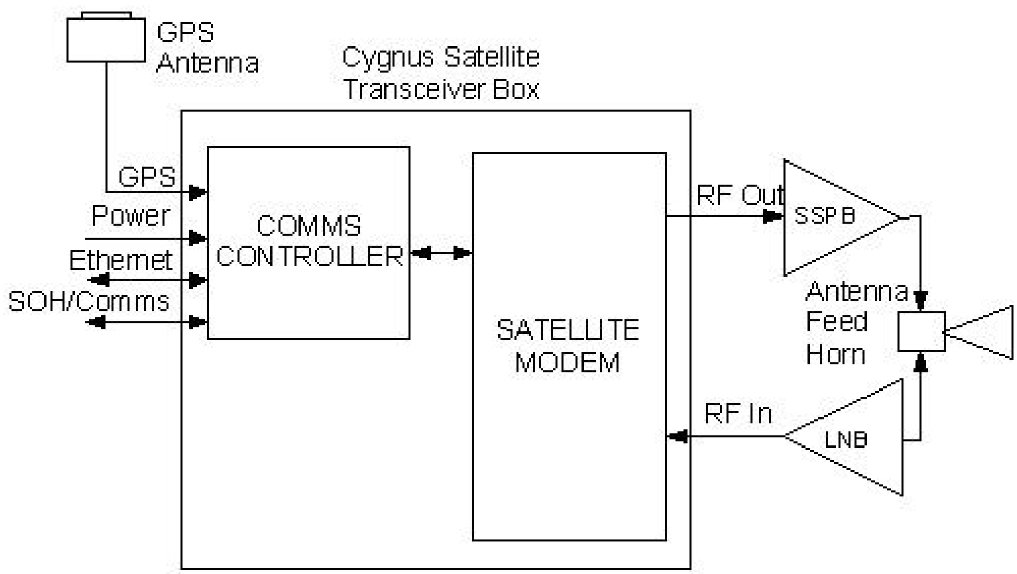


Figure 1: Cygnus satellite transceiver block diagram.

The Cygnus Satellite transceiver is packaged in a rugged waterproof and environmentally resistant steel box. All the connectors are situated on the front plate of the box as shown in the Figure 2.

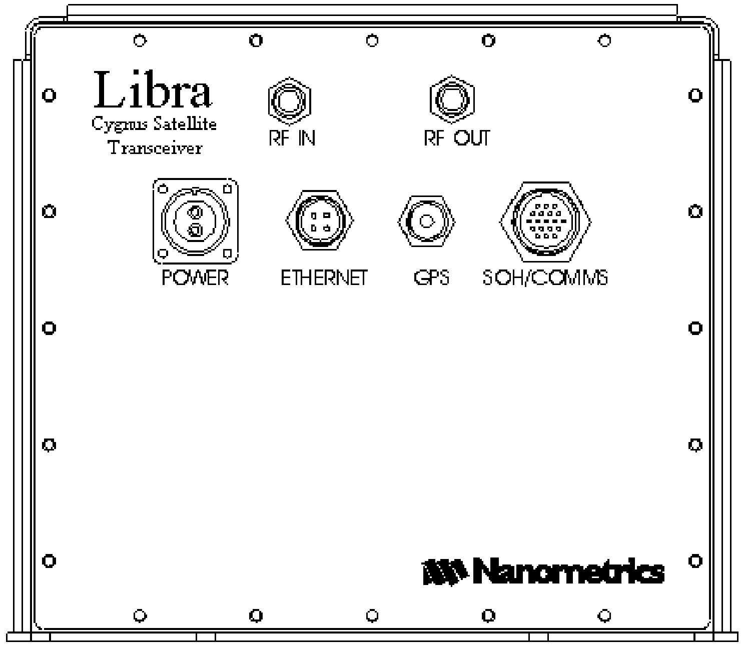


Figure 2: Cygnus Satellite transceiver front plate

Specifications

Supply voltage: 11 - 16 VDC,

Power consumption: 15 to 30 watts (configuration dependent)

Operating temperature: -20 to +55 degrees C

Humidity 0 to 100%

Comms Controller Module

Operation

The communications controller board receives serial data on the two serial ports and prepares the data for transmission over the VSAT link. The packets are then passed to the next module within the Cygnus satellite transceiver - the Satellite Modem.

Each of the serial ports of the communications controller board can be configured using the Libra User Interface program to operate in following modes:

1. Transparent serial port operation: the communications controller packetizes the RS232 data received on the serial port into user datagrams and prepares them for transmission using the UDP/IP protocol over the satellite link. This operational mode should be used when the data generated by a data source should be transmitted to the data acquisition center.
2. Interactive operation: the communications controller uses the TCP/IP protocol for data transfer from the serial port via the satellite link. In interactive mode operators at

the central site can send commands and receive messages from the device connected to the serial port of the communications controller.

3. Console: the communications controller operation can be monitored via the Cygnus test cable (see appendices). In this case information or data received on the serial port is not transmitted over the satellite.

Cygnus supports full duplex satellite communications with both the UDP and TCP protocols. This allows operators to directly communicate via a satellite link with the communications controller and the attached devices. Data received on the serial ports is packetised in NMX packets (NMXP format) and than these packets are embedded into UDP packets for transmission over the satellite link. In addition to being transmitted to the central site each NMXP packet is being stored in the internal memory of the instrument. If the data is not received at the central station by the NAQS network data acquisition software, a request is generated for retransmission. At the receipt of the retransmission request the communications controller fetches the requested packet from its memory and queues it for transmission. The internal memory is a ringbuffer type memory in which the oldest packet is continuously overwritten by the newest one. A detailed description of the NMXP data format can be found in Appendix B.

To minimize the satellite space segment lease cost the Cygnus satellite transceiver uses a powerful implementation of the Time Division Multiple Access (TDMA). This frequency sharing method consists of sequentially ordered time slots. Each transceiver can proceed with transmission only in the time slot assigned to it. The order of the time slots is transmitted to VSAT remote stations by the Carina hub.

To simplify network maintenance the communications controller generates equipment state-of-health messages which are sent to the NAQS Server software (using the UDP protocol) and to the Libra User Interface (suing the TCP protocol).

There are a number of configurable parameters which control the operation of the Cygnus Satellite transceiver. These can be accessed and controlled using a user friendly software package called the Libra User Interface. This software is essentially the graphical user interface of the Nanometrics Libra satellite communication system. It allows access and configuration of each satellite transceiver and hub in the network. For more detailed information on the operation and usage of the Libra User Interface software refer to the separate Libra User Interface manual.

Specifications

Serial ports

Ports: 2

Serial data baud rate: asynchronous 1.2 to 19.2 kbps RS232. Serial ports can be configured for transparent full duplex serial with choice of record termination characters or time-out.

Ethernet port

Ethernet port: 10-based-T Ethernet port. The Ethernet port can be used for on-site configuration or for order wire support (optional).

Precision timing subsystem

Digitally compensated crystal oscillator phase locked to GPS receiver

Eight channel GPS receiver with antenna

Precision UTC reference for system timing

No long-term timing or frequency measurement error

Transmitter shutdown in the absence of GPS signal.

State-of-health channels & Remote site Log

Internal electronics module temperature

Power supply voltage

Time variation (GPS time minus station time) at GPS lock

Phase lock loop status (locked, fine locked, free running)

GPS status (2D Nav, 3D Nav, searching, single satellite)

GPS location, Lat, Long and Elevation.

GPS # of satellites tracked and signal strength

Transceiver carrier & synthesizer lock

Outbound block error rate

Receiver signal to noise ratio

Control

Supports Central Site control of transmit & receive frequency & transmitter power.

Configuration, setup and status are accessible via the satellite link or local Ethernet port.

Responds to remote assignment of TDMA slot and transmit interval and packet length.

Transmitter shutdown on frame time error or self test failure. Self test on startup

Satellite Modem Module

Operation

The Satellite Modem board is a functional part of the Libra remote VSAT transceiver, together with the SSPB and LNB modules mounted on the satellite dish antenna. The transceiver uses an efficient QPSK modulation which allows up to 112 kbps of data to be passed through a single 100 kHz satellite channel. The system is factory configured to operate with standard Ku-band or C-band geostationary communication satellites.

Specifications

Ku band operation, full duplex (other bands optionally available)

Transmit frequency 14.0 to 14.5 GHz

Receive frequency 10.95 to 11.7 Ghz or
11.7 Ghz to 12.2 GHz (factory option) or
12.2 Ghz to 12.75 GHz (factory option)

Carrier tuning step size: 100 Hz

Transmit frequency stability: +/- 2 Khz

Spurious emissions: <4dBW / 4KHz, 14-14.5 GHz

Tx transmit level: 1W max,.5W typical operating, adjustable in 0.25 dB steps by remote control

Idle state emission: -50 dBc

Occupied bandwidth: 90 KHz typical, -26dB at +/- 50 KHz

Modulation: QPSK or BPSK

Data rate: 64 Kbit using rate ½ FEC

Rx demod sensitivity: <10-5 BER at Eb/No=5 dB using ½ rate FEC

6. Field Installation

This section of the manual describes how to configure the hardware and install the Cygnus for field deployment. This section does not include instructions concerning radio tower, antenna, power supply system or any other remote site hardware installation. Information and recommended instructions referring to installation of other remote site hardware can be found in the Hardware Installation Manual.

The Cygnus Satellite transceiver is mounted at the back of the satellite antenna and connected as shown in figure 3

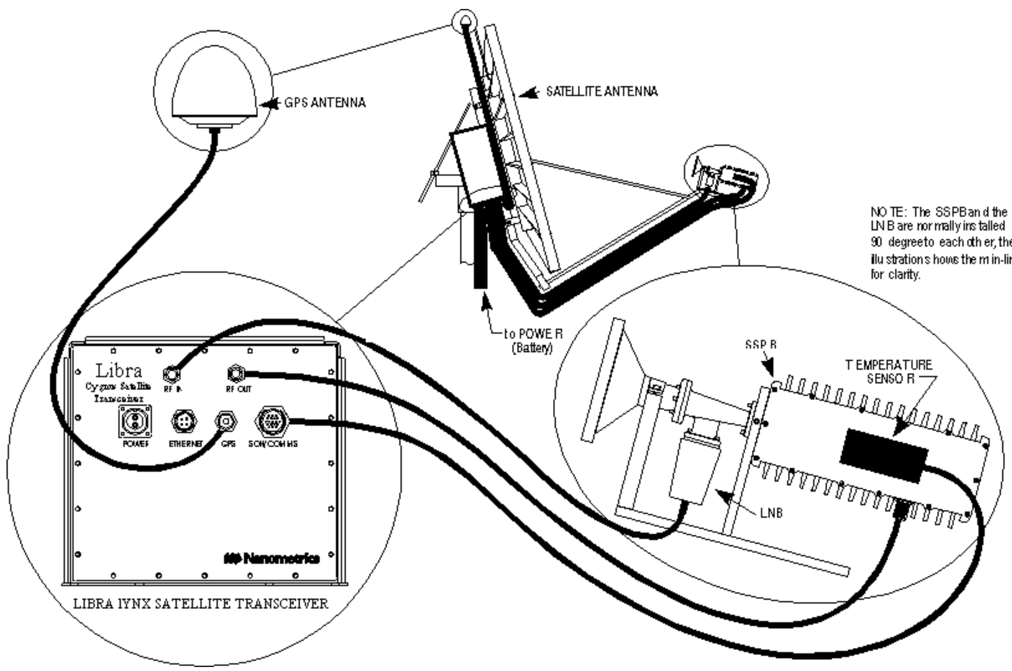


Figure 3: Cygnus Satellite transceiver installation

Cygnus does not require any hardware configuration for remote field deployment. The installation of Cygnus at the remote station consists of mounting the unit on the support hardware and connecting the necessary cables to the unit. The sizes of the front panel connectors are different for each connector and they are provided with polarization control guidance. RF antenna, GPS antenna and power cables are included with the unit. Other cables are either factory supplied or customer built depending on the contract.

For more information on configuring and installing the Cygnus refer to the Remote Station Installation Manual.

Warning: Do not connect the transmit power cable to Cygnus until you confirm that the unit is correctly configured. The right configuration parameters can be read from the Remote Site configuration sheet which is completed up by the Technical Administrator of the system.

.

This page intentionally left blank.

7. Servicing

Maintenance

Repair philosophy

Object is to troubleshoot to the main assembly level and replace the entire Cygnus.

Disassembly Instructions

Disconnect all cables and send the unit back to Nanometrics for repair.

Assembly Instructions

See the Installation chapter of this manual.

Configuration Port

The configuration port is an RS-232 port used for configuration and firmware downloading. It is a three pin port: RX, TX, and GND. The port is on the SOH/communications connector:

TX	Pin L
GND	Pin U
RX	Pin V

Access to that port is enabled with the Cygnus test cable through the connector marked "Controller Config". The test cable can be connected to the SOH/Config connector of Cygnus at one end and to the serial port of a computer at the other end. The Cygnus operation can be monitored from a terminal emulator, such as Hyperterminal reading the COM port of the computer to which the Cygnus test cable is connected. The terminal emulator should be configured for 8 bits, no parity and 1 stop bit (8N1) and no hardware or software handshaking. Although the Cygnus configuration cannot be changed from its configuration port accessing the unit via this port with the test cable and a terminal emulator is very useful in finding out the IP addresses configured in Cygnus.

See Appendix G for detailed pin assignment description of a factory test cable and other cables.

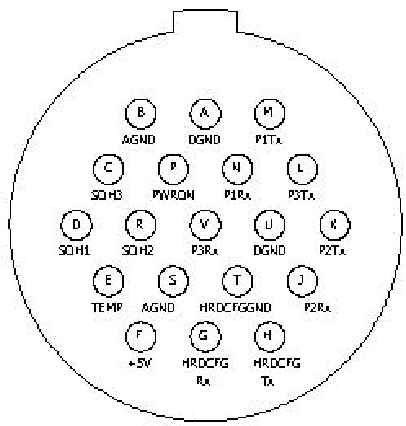
Firmware Updates

New firmware is uploaded through either the Ethernet port or the satellite link using the Libra User Interface program. For more detailed information on this matter consult the Libra User Interface Manual. Firmware update cannot be done via the configuration port using the Cygnus test cable.

NEVER power down the Cygnus while it is doing an upload. This might have catastrophic results.

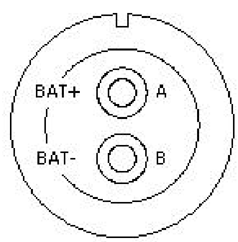
This page intentionally left blank.

Appendix A - Connector Pinouts



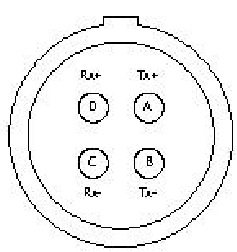
COMMUNICATIONS

SOURIAU 851-07A14-19S-N50



POWER

CANNON CA-02L16-11P



ETHERNET

SOURIAU 851-07A-8-4S-N50-A7

Appendix B - NMXP Data Format

Introduction

Data received on the serial port of the instrument is packetized in NMXP format and then these packets are embedded in standard UDP packets. This data transmission format facilitates the transfer of data along with a wide variety of status information from an instrument to a central site. It supports error free transmission of data using retransmission requests of bad packets. The data format requires that the instrument have an accurate time source (i.e. GPS) for time tagging the data prior to transmission. Most of the status messages can be transmitted at a user defined frequency. This allows the user to tailor the ratio of data to status information. This is important on limited bandwidth or noisy transmission mediums. The status information in data format is expandable. As new status information messages are created, they can be added to the data format without affecting the existing information.

Lastly, the data format is simple to implement on small microprocessors.

The following objectives were used in designing the data format:

1. support retransmit of packets for error correction

2. simple to implement

3. expandable

4. programmable frequency for status information

5. efficient bandwidth usage

Description of Packets

All the data is gathered into sequenced and time stamped packets. These packets start with a synchronization word plus an Old packet available word and finish with a CRC. The packets consist of 17 byte 'bundles' of data. Each bundle is an independent collection of data. Each packet contains a time stamp bundle followed by n data bundles where n is odd. In order to word align packets, an odd number of bundles is used.

The number of bundles in a packet is 19.

In summary:

- outbound data is data being transmitted from the central recording site to the field stations
- inbound data is data being transmitted from the field stations to the central recording site
- a channel is a unique stream of information (i.e.: serial port 1)
- an instrument may transmit 1 or more channels of information
- a packet contains information from only one channel
- a packet is a uniquely identifiable collection of information that is transmitted
- outbound packets do not have the Oldest packet word
- inbound packets contain data, status, or configuration information
- outbound packets contain retransmit requests, or configurations
- outbound is from the central site to remote site, inbound is the opposite
- all data is represented in the little endian format (intel format)

CRC for the Packets

For simplicity CRCs should be an addressable data size, i.e. 8, 16 or 32 bits. Sending 32 bits (4 bytes) is too much overhead, 8 bits is not sufficient. Therefore select 16 bits as the CRC size.

There are 2 good common standard 16 bits polynomials, the CRC-16 and the 16 bit CRC-CCITT. The reversal of these polynomials are also known to be good and are also used. The satellite transceiver uses the 16 bit CRC-CCITT. Therefore select the 16 bit CRC-CCITT as the CRC polynomial.

On the bit level CRCs can be calculated from either end. Normally CRCs are calculated most significant bit first, i.e. the byte 0xD5 is done 11010101. Reflected CRCs are calculated least significant bit first, i.e. the byte 0xD5 is done 10101011. Bits of a byte are transmitted over a serial link least significant bit first. Most CRCs calculated for transmission over a serial link are done least significant bit first. The satellite transceiver uses a reflected CRC algorithm. Therefore, select a reflected CRC algorithm to be used.

If the accumulated CRC value is 0, it is unaffected by the 0 byte and errors may not be detected. The CRC should not be initialized to 0 if messages contain leading 0s may occur. Since the receiver syncs on a non-zero sync word, this is not a problem if these bytes are included. The satellite transceiver uses 0xFFFF as its initial value for the CRC. Therefore, select 0 as the initial value for the CRC.

If the CRC is sent as is, the accumulated CRC value afterwards is then 0. This can pose problems if messages have the CRC and trailing 0s; see above. Modifying the CRC before transmitting it has little to gain and complicates matters. The Orion XORs the CRC with 0xFFFF before transmitting it and checks for 0xF0B8 on receive. Therefore, select transmitting the CRC as is.

```
#define CrcUpdate(usCrc,ubByte) \  ((usCrc) >> 8) ^ ausCrcTable [((usCrc) &
0xff) ^ (ubByte)]
SendByte (ubByte)
{
       usCrc = CrcUpdate (usCrc,ubByte);
       UscTx = ubByte ^ ubScramble;
}
RecvByte ()
{
       ubByte = UscRx ^ ubScramble;
       usCrc = CrcUpdate (usCrc, ubByte);
       return ubByte;
}
SendMsg (pubData)
{
       usCrc = 0;
       SendByte (ubSync1);
       SendByte (ubSync2);
       SendLong (ulOldestSequenceNumber);
       for (us = 0; us < usNumberMsgByte, us ++)
       SendByte (pubData [us]);
       usCrc2 = usCrc;
       SendWord (usCrc2);
}

RecvMsg (pubData)
{
       while (1)
       {
```

```
            while (1)
            {
                while (1)
                {
                    usCrc = 0;
                    if (RecvByte() == ubSync1)
                        break;
                }
                if (RecvByte () == ubSync2)
                    break;
            }
            ulOldestSequenceNumber = RecvLong ();
            for (us = 0; us < usNumberMsgByte, us ++)
                pubData [us] = RecvByte ();
            usCrc2 = usCrc;
            if (usCrc2 == RecvWord () && usCrc == 0)
                break;
        }
}


unsigned short ausCrcTable[256] ={
  0x0000, 0x1189, 0x2312, 0x329B, 0x4624, 0x57AD, 0x6536, 0x74BF, 0x8C48,
0x9DC1,
  0xAF5A, 0xBED3, 0xCA6C, 0xDBE5, 0xE97E, 0xF8F7, 0x1081, 0x0108, 0x3393,
0x221A,
  0x56A5, 0x472C, 0x75B7, 0x643E, 0x9CC9, 0x8D40, 0xBFDB, 0xAE52, 0xDAED,
0xCB64,
  0xF9FF, 0xE876, 0x2102, 0x308B, 0x0210, 0x1399, 0x6726, 0x76AF, 0x4434,
0x55BD,
  0xAD4A, 0xBCC3, 0x8E58, 0x9FD1, 0xEB6E, 0xFAE7, 0xC87C, 0xD9F5, 0x3183,
0x200A,
  0x1291, 0x0318, 0x77A7, 0x662E, 0x54B5, 0x453C, 0xBDCB, 0xAC42, 0x9ED9,
0x8F50,
  0xFBEF, 0xEA66, 0xD8FD, 0xC974, 0x4204, 0x538D, 0x6116, 0x709F, 0x0420,
0x15A9,
  0x2732, 0x36BB, 0xCE4C, 0xDFC5, 0xED5E, 0xFCD7, 0x8868, 0x99E1, 0xAB7A,
0xBAF3,
  0x5285, 0x430C, 0x7197, 0x601E, 0x14A1, 0x0528, 0x37B3, 0x263A, 0xDECD,
0xCF44,
  0xFDDF, 0xEC56, 0x98E9, 0x8960, 0xBBFB, 0xAA72, 0x6306, 0x728F, 0x4014,
0x519D,
  0x2522, 0x34AB, 0x0630, 0x17B9, 0xEF4E, 0xFEC7, 0xCC5C, 0xDDD5, 0xA96A,
0xB8E3,
  0x8A78, 0x9BF1, 0x7387, 0x620E, 0x5095, 0x411C, 0x35A3, 0x242A, 0x16B1,
0x0738,
  0xFFCF, 0xEE46, 0xDCDD, 0xCD54, 0xB9EB, 0xA862, 0x9AF9, 0x8B70, 0x8408,
0x9581,
  0xA71A, 0xB693, 0xC22C, 0xD3A5, 0xE13E, 0xF0B7, 0x0840, 0x19C9, 0x2B52,
0x3ADB,
  0x4E64, 0x5FED, 0x6D76, 0x7CFF, 0x9489, 0x8500, 0xB79B, 0xA612, 0xD2AD,
0xC324,
  0xF1BF, 0xE036, 0x18C1, 0x0948, 0x3BD3, 0x2A5A, 0x5EE5, 0x4F6C, 0x7DF7,
0x6C7E,
  0xA50A, 0xB483, 0x8618, 0x9791, 0xE32E, 0xF2A7, 0xC03C, 0xD1B5, 0x2942,
0x38CB,
  0x0A50, 0x1BD9, 0x6F66, 0x7EEF, 0x4C74, 0x5DFD, 0xB58B, 0xA402, 0x9699,
0x8710,
  0xF3AF, 0xE226, 0xD0BD, 0xC134, 0x39C3, 0x284A, 0x1AD1, 0x0B58, 0x7FE7,
0x6E6E,
```

```
  0x5CF5, 0x4D7C, 0xC60C, 0xD785, 0xE51E, 0xF497, 0x8028, 0x91A1, 0xA33A,
0xB2B3,
  0x4A44, 0x5BCD, 0x6956, 0x78DF, 0x0C60, 0x1DE9, 0x2F72, 0x3EFB, 0xD68D,
0xC704,
  0xF59F, 0xE416, 0x90A9, 0x8120, 0xB3BB, 0xA232, 0x5AC5, 0x4B4C, 0x79D7,
0x685E,
  0x1CE1, 0x0D68, 0x3FF3, 0x2E7A, 0xE70E, 0xF687, 0xC41C, 0xD595, 0xA12A,
0xB0A3,
  0x8238, 0x93B1, 0x6B46, 0x7ACF, 0x4854, 0x59DD, 0x2D62, 0x3CEB, 0x0E70,
0x1FF9,
  0xF78F, 0xE606, 0xD49D, 0xC514, 0xB1AB, 0xA022, 0x92B9, 0x8330, 0x7BC7,
0x6A4E,
  0x58D5, 0x495C, 0x3DE3, 0x2C6A, 0x1EF1, 0x0F78
};
```

Outbound Packets

All outbound packets are a fixed size of 30 bytes. Instruments do not request retransmission of outbound packets.

The basic format is as follows:

2 bytes	Synchronization Word
2 bytes	Instrument ID (5 bits model type, 11 bits serial number)
4 bytes	Long Seconds
1 byte	Packet Type
3 bytes	Packet Information Header
4 x 4 bytes	Data Section
2 bytes	Packet CRC

The 20-bytes type-specific section contains the packet type, information header and data sections. The type-specific section for supported packets are defined below.

Retransmission Request by Sequence Number Range Packet

1 byte	Packet type = 2
1 byte	Channel
1 byte	Data type
1 byte	Spare
4 bytes	First sequence number requested
4 bytes	Last sequence number requested
8 byte	Spare

1. Data type is the data packet type requested (1 = time series, 2 = SOH, transparent serial = 6, etc.).
2. If the data type byte is zero, type can be determined from channel (0-5 = time series, -1 = SOH).

Inbound Packets

All inbound packets consist of a synchronization pattern, oldest packet available for a data stream, time stamp bundle, n other bundles, and a CRC:

2 bytes	Synchronization Word
4 bytes	Oldest Packet Available for a Data Stream
17 bytes	Packet Header
17 * n bytes	n bundles where n is odd
2 bytes	Packet CRC

Packet Header

1 byte	Packet type
4 bytes	Long seconds in seconds since 1970
2 bytes	packet specific
2 bytes	Instrument ID [5 bit model type, 11 bit serial number]
4 bytes	Sequence Number
4 bytes	packet specific

The instrument ID defines the instrument type transmitting the channel of data (CYGNUS = 5)

All inbound packets consist of a synchronization pattern, oldest packet available for a data stream, time stamp bundle, n other bundles, and a CRC:

2 bytes	Synchronization Word
4 bytes	Oldest Packet Available for a Data Stream
17 bytes	Packet Header Bundle
17 * n bytes	n bundles where n is odd
2 bytes	Packet CRC

Transparent Serial Packet

A transparent serial packet contains a time stamp header followed by N data bytes (where N is user defined subject to N = 17 * k, where k is an odd integer, 1 <= k <= 59). k is typically 15, which gives N = 255. The time stamp bundle contains a sequence number, the time of the first sample, instrument ID, channel number, and the number of valid payload bytes, M.

Transparent serial packets are normally sent when the packet is full (M = N). However, the packet sender may be configured to send partial packets after a time out (i.e. if a specified time has passed since the first byte of the packet was received). In this case, M < N, and the last N - M bytes should be discarded. Partial packets are always padded out to full length.

Transparent Serial Packet Format:

1 byte	Packet type = 6 (bit 5 = 1 is for retransmit)
4 bytes	Long seconds
2 bytes	Sub-seconds in 10,000th of a second
2 bytes	Instrument ID [5 bit model type, 11 bit serial number]
4 bytes	Sequence Number
2 bytes	Number of bytes of payload data

1 byte	Channel number (port number)
1 byte	spare
N bytes	binary serial data

State-of-Health Packets

A state-of-health packet consists of a status time stamp bundle followed by n status bundles. A status time stamp consists of a sequence number, the time (nominal time when the packet was created), instrument ID (model and serial number).

Status bundles have a general format that is outlined below:

1 byte	bundle type = xx
4 bytes	Long seconds
12 bytes	Defined by the specific bundle type

Status Packet Header Bundle

1 byte	Packet type = 2 (bit 5 = 1 is for retransmit)
4 bytes	Long seconds
2 bytes	Sub-seconds in 10,000th of a second, this value always 0
2 bytes	Instrument ID [5 bit model type, 11 bit serial number]
4 bytes	Sequence Number
1 byte	indicates test packet if (byte & 0x01 != 0)
3 bytes	Reserved for future use

Null Bundle

This bundle is provided to pad out packets. The first occurrence of a Null bundle indicates that there is no further data in the packet. The null bundle contains no useful information. The receiver should disregard this bundle and may skip to the next packet.

1 byte	Bundle Type = 9
16 bytes	Filler

GPS Location Bundle

This bundle contains the latitude and longitude of the instrument's GPS antenna. This bundle has a programmable measurement frequency. The latitude and longitude is stored in IEEE floating point format.

1 byte	Bundle type = 13
4 bytes	Long seconds
4 bytes	Latitude
4 bytes	Longitude
4 bytes	Elevation

GPS Satellite Information Bundle

1 byte	bundle type = 40
4 bytes	long seconds
1 byte	MillisecFlag | Channel #
	bits 0-3 Channel # (0-15)

bits 4-7 Millisec Flag

1 msec from sub_frame data collection

2 verified by a bit crossing time

3 verified by successful position fix

4 suspected msec error

1 byte	Acquisition Flag | PRN	
	bits 0-4 PRN	
	bits 5-7 Acquisition Flag:	
		0 = unlocked
		1 = search
		2 = track
1 byte	Elevation (0-255): el= value/255x90	
1 byte	Azimuth (0-255: az = value/255x360	
2 bytes	Signal Level	
6 bytes	repeat for another channel - see the 6 bytes above	

Serial Port Map Bundle

1 byte	int8	bundle type = 41
4 bytes	long	long seconds
1 bytes	int8	index
1 bytes	int8	serial port number
2 bytes	int16	number of minutes since last packet arrived
2 bytes	int16	HRD instrument ID (see data packets)
6 bytes	-	spare

Telemetry Packet Reader Errors Bundle

1 byte	int8	bundle type = 42
4 bytes	long	long seconds
1 bytes	int8	serial port number
3 bytes	int24	Bad Packets since startup or start of the day
3 bytes	int24	Good Packets since startup or start of the day
3 bytes	int24	Lost Packets since startup or start of the day
2 bytes	int16	Tx Packets sent by Naqs since startup or start of the day

Serial Port Errors Bundle

1 byte	int8	bundle type = 43
4 bytes	long	long seconds
1 bytes	int8	serial port number
4 bytes	long	serial port overrun errors since startup or last reboot (continuously increases, then wraps, it is never zeroed)
4 bytes	long	serial port frame errors since startup or last reboot (continuously increases, then wraps, it is never zeroed)
3 bytes	spare	

Receiver Slot State Bundle

1 byte	int8	bundle type = 44
4 bytes	long	long seconds
4 bytes	int32	receiver IP address
2 bytes	int16	DQT_AGC - AGC level for quadrature tuner in units of 0.1 dB
2 bytes	int16	carrier offset in units of 10 Hz
2 bytes	int16	symbol offset in Hz
1 byte	int8	DCL_AGC - AGC level for Costas loop in units of 0.1 dB
1 byte	-	spare

Transmitter Slot Error Bundle

1 byte	int8	bundle type = 45
4 bytes	long	long seconds
4 bytes	int32	transmitter IP address
4 bytes	int32	no. of bad packets since the start of this TDMA configuration
4 bytes	int32	no. of good packets since the start of this TDMa configuration

Receiver Slot Error Bundle

1 byte	int8	bundle type = 47
4 bytes	long	long seconds
4 bytes	int32	receiver IP address
4 bytes	int32	no. of bad packets since the start of this TDMA configuration
4 bytes	int 32	no. of good packets since the start of this TDMA configuration

Libra Instrument SOH Bundle

1 byte	int8	bundle type = 48
4 bytes	long	long secondes
2 bytes	int16	ten MHz frequency error
2 bytes	float16	SSPB temperature
2 bytes	float16	WW temperature
2 bytes	float16	TX temperature
2 bytes	float16	battery temperature
2 bytes	-	spare

Libra Environment SOH Bundle

1 byte	int8	bundle type = 49
4 bytes	long	long seconds
4 bytes	float	external SOH channel 1 (scaled)
4 bytes	float	external SOH channel 2
4 bytes	float	external SOH channel 3

Transmitter Bundle

1 byte	int8	bundle type = 50
4 bytes	long	long seconds

4 bytes	int32	transmitter IP address
4 bytes	int32	transmitter frequency in HHz
4 bytes	int32	transmitter level

Receiver Bundle

1 byte	int8	bundle type = 51
4 bytes	long	long seconds
4 bytes	int32	receiver IP address
4 bytes	int32	receiver frequency in Hhz
4 bytes	-	spare

Burst Bundle

1 byte	int8	bundle type = 52
4 bytes	long	long seconds
4 bytes	int32	transmitter IP address
1 byte	int8	bits 0-1: slot state 0 = find (sweeping for carrier) 1 = verify (has carrier, looking for data) 2 = track (receiving data) bits 2-3: burst state for most recent burst 0 = not found 1 = found CW 2 = found UW 3 = found data
3 bytes	int24	no. of good burst since the start of this TDMA configuration
3 bytes	int24	no. of bad burst since the start of this TDMA configuration
1 byte	-	spare

Epoch Bundle

1 byte	int8	bundle type = 53
4 bytes	long	long seconds
4 bytes	int32	next epoch start time (seconds since 1970)
8 bytes	spare	

Libra GPS Time Quality Bundle

1 byte	int8	bundle type = 54
4 bytes	long	long seconds
2 byte	short	GPS status 0: computing position fixes (navigating) 1: no_time 2: needs initializing 3: pdop_too_high (no solution) 8 to 11: acquiring (8 + #satellites tracked))
2 bytes	short	number of usable satellites

4 bytes	int32	PDOP value
4 bytes	int32	TDOP value

Libra System Time Quality Bundle

1 byte	int8	bundle type = 55
4 bytes	long	long seconds
4 bytes	int32	system time quality: -10: time_unknown -1: time_not_good n >= 0: worst prediction of time error in nsec
2 bytes	int16	PLL mode 1: fine_lock 2: coarse_lock 3: no_lock)
2 bytes	int16	time displacement (system time - GPS time in nanoseconds)
2 bytes	int16	time velocity
2 bytes	float16	current compensation

Libra Operation State Bundle

1 byte	int8	bundle type = 56
4 bytes	long	long seconds
4 bytes	int32	bitfield indicating operating state: bit 0 (LSB): network transmission state: on = 1, off = 0 bits 1-31: reserved for future use
8 bytes	spare	

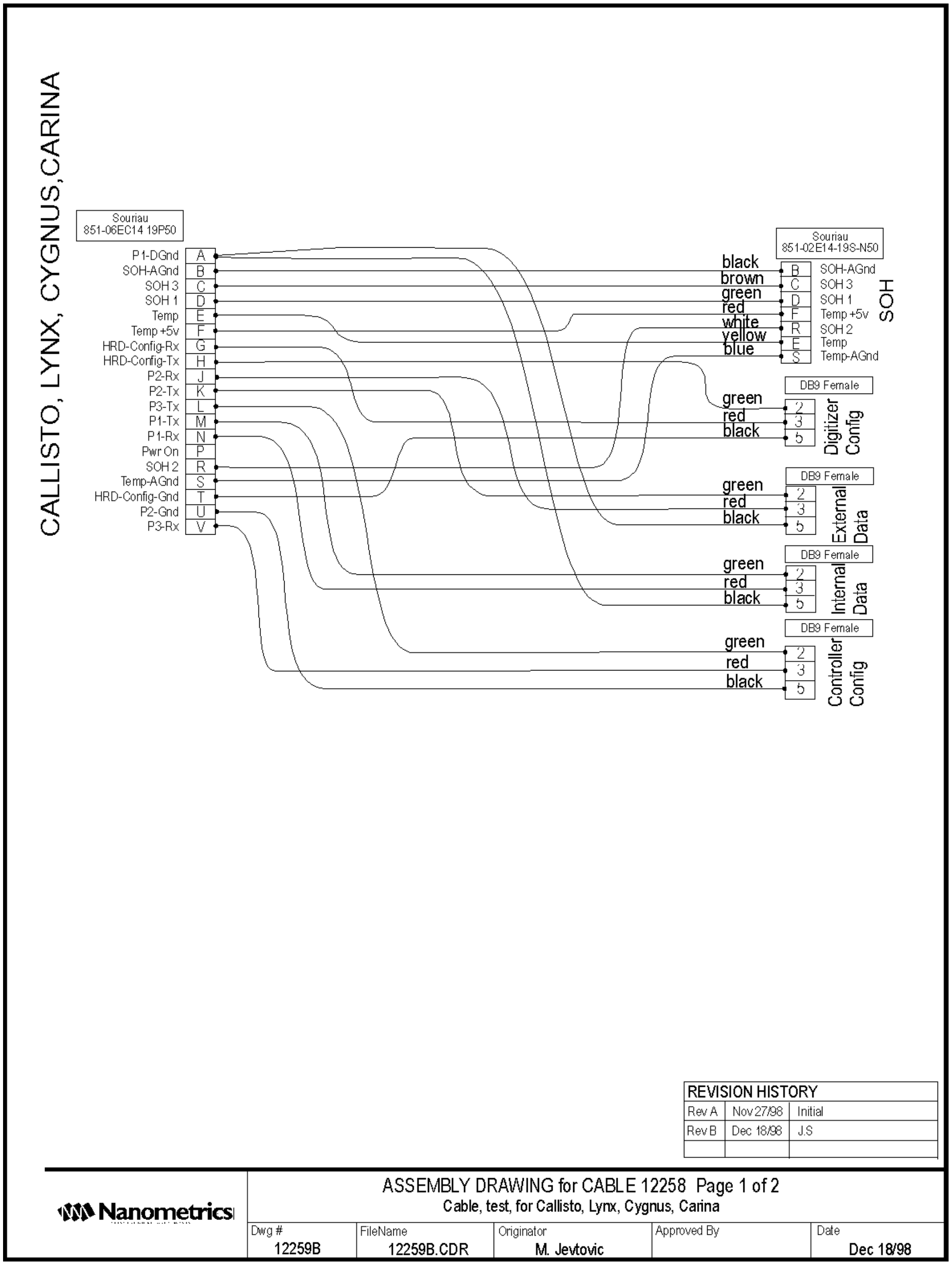
Serial Data Bytes Bundle

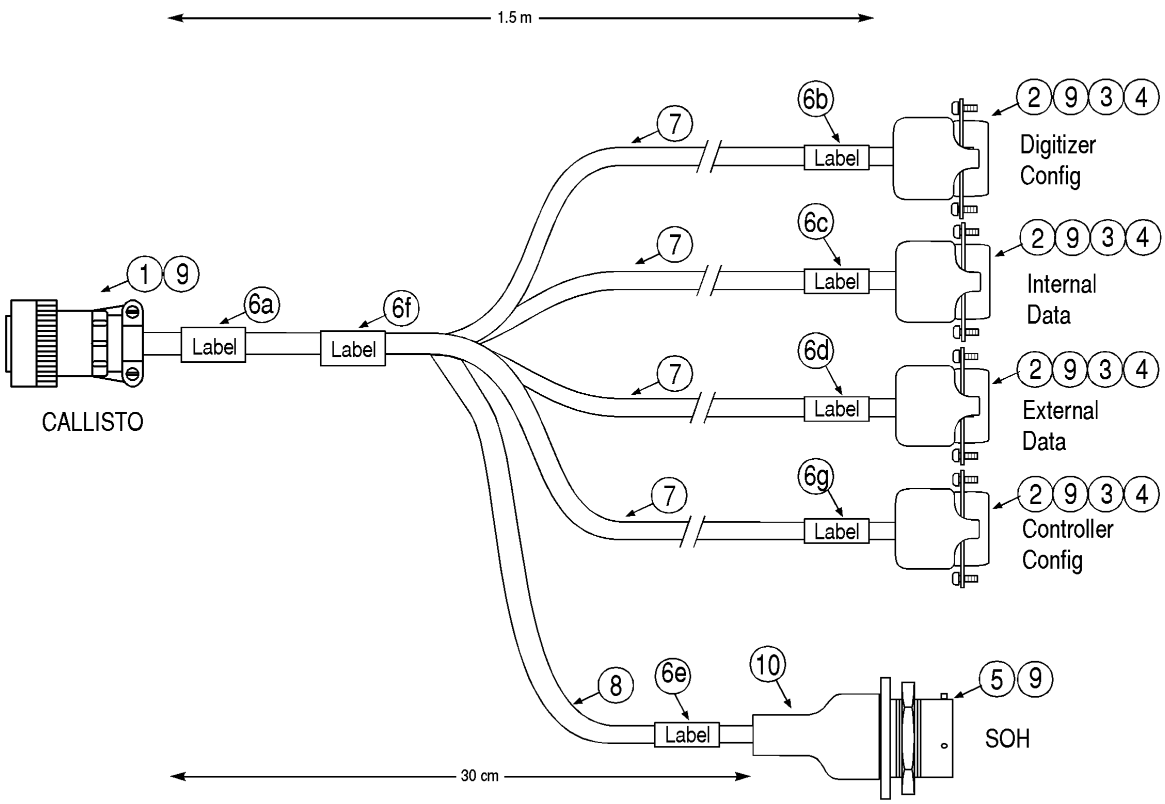
1 byte	int8	bundle type = 57
4 bytes	long	long seconds
1 byte	int8	port number
4 bytes	int32	bytes read since the startup or the start of the day
4 bytes	int32	bytes written since the startup or the start of the day
3 bytes	spare	

.

Appendix C - External Cable Drawings

Important note*: The following pictures are the cable drawings for the test cable used with a set of different type of Nanometrics hardware, such as Callisto digitiser for RF networks, Lynx satellite digitiser and Cygnus satellite transceiver.*





NOTES:

1) Heat shrink tubing to be used on all soldered connector.

2) Heavy wall heat shrink tubing (10) to be used on connector (5).

REVISION HISTORY		
Rev A	Nov 27/98	Initial
Rev B	Dec 18/98	J.S.

Nanometrics

ASSEMBLY DRAWING for CABLE 12258 Page 1 of 2

Cable, test, for Callisto

Dwg #	FileName	Originator	Approved By	Date
12259B	12259B.CDR	M. Jevtovic		Dec 18/98