Sensor Link Protocol:

A Common Digital Information Link for Sensor Systems

31 March, 1998

Bill Peters

Program Management Office for Night Vision / Reconnaissance Surveillance and Target Acquisition
U.S. Army - PM NV/RSTA
Ft. Belvoir, VA. 22060

James Meehan, Dennis Miller, Doug Moore

Nichols Research Corporation
4040 South Memorial Parkway
Huntsville, AL. 35801

ABSTRACT

The Program Management Office of Night Vision/Reconnaissance Surveillance and Target Acquisition (NV/RSTA) has developed the Sensor Link Protocol which permits a “plug n play” like integration of a diverse set of sensors currently or soon to be in production. The Sensor Link Protocol is an RS 485/232 based networking protocol, which allows a variety of sensor systems to be connected to a diverse set of computer platforms. The protocol then provides an interface through which digital information can be passed between the host computer and the sensor as well as a method of externally controlling the sensor functions.

The continued emphasis on battlefield digitization and communications has created a means to disseminate accurate and timely information among a variety of battlefield computer systems. These efforts now require the digitally interfacing of Reconnaissance, Surveillance and Target Acquisition (RSTA) sensor systems to these battlefield computer systems. This paper describes and outlines the Sensor Link Protocol which provides a common interface to a variety of RSTA sensor systems. The Sensor Link Protocol acts as an enabling technology linking RSTA sensor systems to the digitized battlefield.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (FROM - TO)
31-03-1998	Conference Proceedings	xx-xx-1998 to xx-xx-1998

4. TITLE AND SUBTITLE
Sensor Link Protocol: A Common Digital Information Link for Sensor Systems
Unclassified

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Peters, Bill ;
Meehan, James ;
Miller, Dennis ;
Moore, Doug ;

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME AND ADDRESS
Program Management Office for Night Vision / Reconnaissance Surveillance and Target Acquisition
U.S. Army - PM NV/RSTA
Ft. Belvoir, VA22060

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS
Director, CECOM RDEC
Night Vision and Electronic Sensors Directorate, Security Team
10221 Burbeck Road
Ft. Belvoir, VA22060-5806

10. SPONSOR/MONITOR'S ACRONYM(S)
11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APUBLIC RELEASE
,

13. SUPPLEMENTARY NOTES
See Also ADM201041, 1998 IRIS Proceedings on CD-ROM.

14. ABSTRACT
The Program Management Office of Night Vision/Reconnaissance Surveillance and Target Acquisition (NV/RSTA) has developed the Sensor Link Protocol which permits a ?plug n play? like integration of a diverse set of sensors currently or soon to be in production. The Sensor Link Protocol is an RS 485/232 based networking protocol, which allows a variety of sensor systems to be connected to a diverse set of computer platforms. The protocol then provides an interface through which digital information can be passed between the host computer and the sensor as well as a method of externally controlling the sensor functions. The continued emphasis on battlefield digitization and communications has created a means to disseminate accurate and timely information among a variety of battlefield computer systems. These efforts now require the digitally interfacing of Reconnaissance, Surveillance and Target Acquisition (RSTA) sensor systems to these battlefield computer systems. This paper describes and outlines the Sensor Link Protocol which provides a common interface to a variety of RSTA sensor systems. The Sensor Link Protocol acts as an enabling technology linking RSTA sensor systems to the digitized battlefield.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	Public Release	11	Fenster, Lynn lfenster@dtic.mil
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER International Area Code Area Code Telephone Number 703767-9007 DSN 427-9007

1. Introduction

As today's armed forces move toward battlefield digitization, the need for an accurate and timely source situational awareness and targeting data has become critical. Much of the emphasis and effort to date has been placed on the interfaces between tactical communications systems, tactical internet protocols, and packet switched network interfaces. Interoperability standards for these networks have been defined and much analysis supporting these standards has been performed.

The interface between these battlefield computer systems and the sensor systems has, however, been neglected to date. In fact, these interfaces have been specifically excluded from interoperability standards. The scope of the information transfer sections of the Joint Technical Architecture (JTA) and the JTA-Army clearly state: "This section identifies standards that support the transfer of data, video, imagery, and multimedia. The standards described in this section apply at the external interfaces between computer systems (i.e., hosts), routers, and communications networks. These standards do not apply at the interfaces between hosts and peripherals (e.g., storage devices, sensors, and weapons control)."

The Program Manager, Night Vision/Reconnaissance, Surveillance and Target Acquisition (PM NV/RSTA) has developed a sensor interoperability architecture that directly addresses the need for standardization in this critical area. This architecture is applicable to man portable, vehicle mounted, airborne, and unmanned or remote sensor Applications. The architecture extends the concepts outlined in the JTA and JTA-A to the interfaces between Hosts and Sensor systems. This concept has been adopted by the Program Executive Offices for Intelligence, Electronic Warfare & Sensors. (PEO IEW&S)

A primary focus of this effort has been the development of the Sensor Link Protocol. The Sensor Link Protocol is an RS 485/232 based networking protocol, which allows a variety of sensor systems to be connected to host computer systems. The Sensor Link Protocol permits a "plug and play" like integration of a diverse set of sensors currently or soon to be in production.

The recent emphasis on the digitization of the battlefield has yielded a new generation of sensors, which have a number of common functions to include precise self-location, precise target location, laser target designation, image transmission, and data dissemination. Prior to the development of the Sensor Link Protocol, each sensor developer created its own unique interface to the various computers platforms available. Only interfaces to devices that the contractor was specifically tasked to support we developed. With the advent of the Sensor Link Protocol, a common interface and common integration capabilities are now available for sensor systems as well as for host computer platforms supporting a variety of operating systems. The interface commonality provided by the Sensor Link Protocol allows direct software reuse at both the host and embedded system level. The layer software components developed for the physical, data link and network layers of the protocol can be directly applied across sensor development programs and various mission applications. The U.S. Army/USMC Lightweight Laser Designator Rangefinder Program (LLDR), the Enhanced Target Location and Observation System (ETLOS), the Lightweight Video Reconnaissance System (LVRS), the Mini Eyesafe Laser Infrared Observation Set (MELIOS), and other candidate systems now incorporate this "plug and play" like interface to the digital Battlefield. Also, this architecture permits the straightforward insertion of experimental devices into existing fielded systems for the purpose of evaluating advanced concepts.

This protocol has potential application well beyond PM, NV/RSTA sensor systems. Actions are currently underway, with the support of the PEO, IEW&S, to submit the protocol for release as a commercial standard, and ultimately, for incorporation into the JTA-A. This would give materiel developers a stable baseline for the development and production of sensor system interfaces.

Device level interface software, which will support the integration of Sensor Link Protocol interfaces into mission specific software applications, is being developed. . This set of device drivers for the Sensor Link Protocol interface will support the integration of Sensor Link Protocol compliant sensor systems into applications running on a variety of host computers under several operating systems. Operating systems supported are scheduled to include UNIX, Windows 95, Windows NT, Windows CE, LYNX real time UNIX, SCO UNIX and MS-DOS. The initial applications of the Sensor Link Protocol interface are being used to support the integration of LLDR into the Marine Corps' Target Location Designation and Hand-off System (TLDHS); the development of an interface between LLDR and the Army's Hand-held Terminal Unit (HTU); and the integration of LLDR into the Army's STRIKER vehicle system.

The overriding benefit of using the Sensor Link Protocol and its associated device driver to interface to the sensor systems lies in the fact that this is a common interface protocol. A computer system using a Sensor Link Protocol device interface will not only be able to communicate with a specific sensor system to perform mission functions, but will be able to integrate with any Sensor Link Protocol compliant sensor system without interface code modifications.

2. Application

The combination of the interface protocol and device-level software modules provides the enabling technology, which allows sensor systems to be incorporated into diverse mission applications. This capability makes the sensor immediately available to a much broader operational community. For example, the original role of the LLDR was to provide precise target location and laser designation capabilities to the Army's dismounted Fire Support Teams. The incorporation of the protocol has greatly facilitated its adoption as the sensor system for mounted Fire Support Teams (STRIKER). The Marine Corps Forward Observer/Forward Air Controller teams will use these same sensor capabilities.

The rapid and responsive digital information flow from the sensor provided by the protocol can be used by tactical platforms across the battlefield. The Army currently plans to use the targeting information to direct artillery fires, and designate for laser-guided munitions and helicopter-borne missiles, such as Hellfire. The Marine Corps, which relies on its fixed-wing assets, (e.g., FA-18) for close air support, will digitally pass the targeting information directly to Marine Aviation. for target engagement by either standard or laser-guided munitions. The digital targeting information and laser designation capabilities provided by the LLDR are not bound by these specific mission applications. The rapid and common integration capabilities provided by the protocol and associated software modules support the LLDR's use in a virtually limitless variety of future operational scenarios. Examples include unmanned ground and aerial vehicles, remote surveillance and/or target engagement systems, and joint precision targeting missions.

In addition to the obvious utility of the protocol and software modules to sensor systems currently under development or entering production, the protocol can also be applied to legacy systems. An example of this kind of application of the protocol to fielded systems is the development of the Digitized MELIOS.

Currently the MELIOS has, as many sensor systems currently do, a contractor designed proprietary interface. In MELIOS's case the interface was designed solely to support system testing and was never intended to be a tactical data interface. Using this interface control of the device is only supported by physically grounding discrete pins together on the test port connector. Thus the MELIOS system as it currently stands cannot be controlled by a serial device and a MELIOS specific interface must be designed by each and every system integrator who wishes to use MELIOS as part of their system.

A solution to this problem, which now allows MELIOS to be directly integrated into the digitized battlefield, has been developed. A small low-cost device which interfaces to the MELIOS test port and provides Sensor Link Protocol compliant RS-232 and RS-485 connections has been developed. This retrofit device can be used to provide a digital interface to any currently fielded MELIOS.

Litton Laser System Division has developed a ruggedized version of this retrofit interface and is now offering it in combination with or as an upgrade to its MELIOS systems. The combination of the MELIOS and its Sensor Link Protocol compliant interface is being referred to as the Digitized MELIOS. This Digitized MELIOS then provides a common and stable interface to which application developers can integrate. In addition, the device level software modules described above can now be used interface to the MELIOS as well as any other Sensor Link Protocol compliant sensor system.

3. Protocol Definition

All data transferred across the Sensor-Host serial data interface are formatted as messages. Messages contain a header portion and may or may not contain a data portion. All header - only messages are referred to as commands; however, some commands may also have data portions. The header portion contains five header words plus one header checksum word (message words 1 through 6). The data portion may contain a maximum of 100 data words (message words 7 through 6+N, where $1 \leq N \leq 100$) plus one data checksum word. The maximum number of words a message may contain is 107, consisting of 6 header words and 101 data elements. If no data portion is sent, no data checksum word is sent.

Message words consist of 16 bits, or two 8-bit bytes. A byte is transferred across the RS-485 interface proceeded by a start bit and followed by a stop bit. No parity bit is used. Words are transferred with the Least Significant Byte (LSB) first, followed by the Most Significant Byte (MSB). Integer and floating point data types consisting of multiple words are transferred starting with the lowest numbered word to the highest numbered word. Bytes are transferred with the least significant bit first.

3.1. Header Word 1

Header word one is used for frame synchronization. For most mission applications the host will be required to integrate to a Precision Lightweight GPS Receiver (PLGR) as well to sensor devices. In order to facilitate integration of the sensor components into mission applications a unique frame sync sequence has been defined. The frame sync is similar in form to the PLGR frame sync but the specific byte pattern has been chosen to distinguish a Sensor Link Protocol interface message from a PLGR message. Because of this unique frame syncing both devices (PLGR and sensor), and both protocols (PLGR and Sensor Link Protocol), can now be integrated using a common physical medium. The unique Sensor Link Protocol Frame Sync consists of the value 249 decimal (F9 hexadecimal, “11111001” binary) in the first byte, followed by the value 135 decimal (87 hexadecimal, “10000111” binary) in the second byte.

3.2. Header Word 2

Header word two contains the Numerical Identification number (NID) for a particular message. Legal values are 0 to 65,535. For example, the Absolute Target Position message, used by the sensor to report target position, is identified by selecting a NID of 5000. A basic description of the generic Sensor Link Protocol message set is provided later in this paper.

3.3. Header Word 3

Header word three defines the number of words contained in the data portion of a message (not including the data checksum word). Legal values are 0 to 100 a value of 0 indicates a header-only message.

3.4. Header Word 4

Header word four is the address field of a message. The first byte contains the destination address and the second byte contains the source address. Destination addresses may include individual unit addresses, group addresses and the universal broadcast address. Each address value is six bits in length The upper-most bit of the source address byte is set when the address was pre-assigned as part of the unit's initial configuration and is not set when the address was assigned by the net controller or is the sign-on address. The remaining bit of the source address byte and the upper two bits of the destination address byte are reserved and unused.

3.5. Header Word 5

Header word five is a 16-bit field containing protocol and message related flags. A logic 1 indicates that the flag is "set". Bit 0 represents the least significant bit and bit 15 represents the most significant bit.

3.6. Header Word 6

Header word six contains a 16-bit checksum used to validate the header portion of the message. The checksum is computed by summing (modulo 2^{16}) the set bits contained in header word one through four, and then performing two's complement on the results.

3.7. Data Words

The message data portion words are completely transparent to the Sensor-Host serial interface protocol and have no restrictions on bit patterns or character groupings. The number of words in the data portion is specified in header word three. This portion does not exist when the value specified in header word three is zero.

3.8. Data Checksum Word

The data checksum word contains a 16-bit checksum used to validate the data portion of the message. The checksum is computed by summing (modulo 2^{16}) the set bits contained in data portion words, and then performing 2's complement on the result. It is always transmitted as the last byte of any message containing a data portion and is not transmitted for header-only messages.

3.9. Message Handshaking

Both the Host and the Sensor may send messages to individual unit addresses requiring acknowledgment. All commands to individual unit addresses are sent requiring acknowledgment. No messages broadcast or sent to group addresses shall require or request acknowledgment. Response to the acknowledgement request can be in the form of either an Acknowledge (ACK) or a Negative Acknowledge (NAK) message.

A message is sent with acknowledgment requested by setting the Acknowledgment Request flag (bit 12 in header word 5). Any message not requiring acknowledgment is considered complete by the transmitting device as soon as it has been sent. The receiving device considers a message not requiring acknowledgment complete when it has been received successfully. If the message is received in error the receiving device ignores it. Any message transmission requiring an acknowledgment is not considered complete by the transmitting device until a message acknowledgment is received. The transmitting device allows for at least one re-transmission of a message that is not acknowledged or is a negative acknowledged. The transmitting device has only a single message awaiting acknowledgment at any time.

In addition to requiring acknowledgment, some commands and messages require other handshaking be performed before the command or message is considered complete. This handshaking is accomplished by setting the Handshake Request bit in header word 5 of the command and by passing an Accept/ Reject message generated by the commanded device. Additionally, an Accept/Reject message may be generated in response to a command that does not have the Handshake Request bit set if required by the commanded device.

A command may be rejected because the received message NID or data is invalid or the receiving device is not in an appropriate mode to process the command. An Accept message may be generated in response to a command that does not have the Handshake Request flag set if the commanded device must first perform some processing before the requested data is available. An Accept/Reject message is output after the output of an ACK for the same command. An Accept/Reject message is not output after the output of a NAK.

Any message transmission requiring message handshaking is not considered complete by the transmitting device until a message accept or reject is received. The transmitting device has only a single message awaiting acceptance at any time. The Host must receive the Accept/Reject within 1.5 seconds of transmitting the original message. Accept/Reject messages are not requested or generated for messages sent to group or broadcast addresses.

3.10. Command Messages

Header only commands instruct the receiving device to perform some activity. Both the Host and the Sensor can issue header only commands. Types of header only messages include connect, disconnect and request commands. The type of command sent I determined by the flags set in header word 5. A connect command is used to request the repetitive output of a message. A disconnect command is used to stop the repetitive output of a connected message. A request command is used to request the one time output of a message. The Host must receive the requested message within 2 seconds of transmitting the request.

4. Networking

In order to support applications where several sensor systems will be integrated, either to share data with each other or to allow a single host computer or integrating device to collect an fuse the data from all the sensors, the Sensor Link Protocol has been defined as a networking protocol. A network of sensors is not actually required to support the fusion of multiple sensor systems but it is highly desirable. For example, all of the sensors providing data for the system could be interface to a single communications port instead of requiring an individual port for each sensor system. The sensors in the network could then be collected into logical groups. Messages, which need to be sent to all of the sensors in the group, could then be sent to the group address with a minimum of overhead. Perhaps the most

important benefit of a sensor network is that the individual sensors would then be capable of sharing data with each other without requiring interdiction of a host computer or integrating device to properly route the data.

One example of such a network would come from the use of LLDR in a vehicle mounted configuration. When mounted to a vehicle, the internal flux gate compass, which LLDR relies on to determine target azimuth and elevation, becomes unstable. Also the information from the internal GPS system is probably less reliable than the information that can be provided by the vehicles Inertial Navigation System (INS). The networking capabilities provided by the protocol allow LLDR to receive information from an external azimuth/elevation device and/or from the vehicles INS without requiring a host computer to perform routing functions which might cause significant data latency.

While the discrete digital IO lines and the analog video signals are not addressable, the serial data interface can be implemented as an addressable network of multiple units. In the case where the interface is implemented as RS-232, the network consists of only two units, the Host and a single Sensor. For RS-485, the interface can support the networking of from 1 to 29 Sensors with the Host, for a total of up to 30 networked units. The Host is the default network controller (NC) for network configurations requiring network control functions. For RS-232, the interface includes separate data receive and data transmit signals, and operates in a full duplex mode. For RS-485, the interface includes a single differential data signal pair, which provides both the data receive and data transmit capabilities. As such, an RS-485 implementation provides the capability to enable and disable the serial data transmitter, while the serial data receiver, Data Terminal Ready (DTR), is always enabled. This implementation of RS-485 is commonly referred to as “2-Wire DTR with Echo”.

Each network unit is assigned a unique individual network address. Multiple units may also be assigned to network groups, so that a single message may be processed by more than a single unit. Each message that is sent through the serial data interface includes addressing information in the header which indicates the message’s destination(s) and source. The destination address for a message can be the address of a single unit, the address of a single network group, or the universal broadcast address. The universal broadcast address indicates that all network units should process the message.

The NC is responsible for assigning individual network addresses to Sensors as they join the network, if they do not already have a network address assigned. Alternatively, a unit may have a default network address pre-assigned as part of its initialization data. In addition, Sensors can be assigned to up to four (4) group addresses. The entire network can be addressed for broadcast messages using the universal broadcast address. The NC has special network responsibilities, to include the assignment of individual unit addresses and group addresses to units as they join the network. Optionally, the network may function without a NC if all units participating on the network are assigned unique network addresses as part of their initialization process.

The NC is always assigned a network address of 1. Upon power up, all other network units not having a default address assigned will use an address of 2, the network sign-on address. As units join the network, the NC sequentially assigns each unit an individual unit address in the range of 3 to 31. Group addresses are in the range of 32 to 62. Address 63 is the universal broadcast address. Units will only process messages that contain either the unit’s individual address, the address of a group to which the unit is assigned, or the universal broadcast address.

Before a unit is assigned an individual unit address from the NC, the unit will process all Unit Address Assignment and Unit Sign-on Reject messages received with the sign-on address as the destination address. A unit using a pre-assigned address will set the pre-assigned address bit in the source address byte of the header address word for all messages it transmits. The NC address will be considered

a pre-assigned address. Only an individual unit address shall be used as the source address for any message.

5. Collision Avoidance

In an RS-485 network, all units on the network share the same data transmit/receive lines. If two or more units attempt to transmit messages simultaneously, the transmitted messages would “collide” on the network, and none of the messages would be received properly by the destination units. A collision management scheme is required to limit and, when they occur, recover from message collisions. This involves avoiding collisions as much as possible, detecting when they occur, and recovering from them when they do occur. The collision management scheme employed for this protocol includes each of these elements.

When any unit on the network has a message to transmit, it will first determine if the network is busy. If the unit detects that the network is busy, it will not attempt to transmit until it detects that the network is no longer busy, or is idle. When the unit detects that the network is idle, the unit will enable its serial data transmitter and transmit its message. At the beginning of the transmission, the unit will start a data receive timer which expires at the end of a period during which all of the transmitted data should have been received by itself and all other network units. Upon transmission of the last byte of the message, the unit will disable its transmitter.

After the transmission is complete, the unit will compare the message sent with what was received while it was in the transmit state. This can be accomplished by comparing the header checksums and, if any, the data checksums of the message sent and the message received. If the checksums are equivalent, the transmission will be considered successful. If they are not, or if the amount of data received when the data receive timer expires is less than that transmitted, the unit will conclude that the message it transmitted collided with a message transmitted by another network unit.

If a collision is detected or the network was busy when the unit attempted to transmit it will wait a random period of time not less than 200ms and not greater than 2200 ms for the network to become idle. At the end of this period, the unit will again attempt to transmit the message, first checking whether or not the network is busy. This sequence will continue until the unit has attempted to send the message a total of three (3) times. If the last attempt is unsuccessful, the unit will indicate failure for the message transmission

6. Message Set

A generic message set has been defined for the Sensor Link Protocol, which allows application of the protocol to a wide variety of sensor systems. The common thread which links the sensor programs that plan to incorporate the protocol is that they all perform target, or feature, location functions. Whether this function is provided through absolute position, relative position or range to target, all of the sensors are at some level involved in the targeting process. It is precisely this targeting information which the host computer must then pass along to higher echelon by whatever communications link it has at its disposal. The message set for the common device protocol and the protocol itself, therefore, provide a generic and necessary link between the sensors performing the targeting function and the tactical communications systems which are hungry for this situational awareness information.

The message set also provides system integrators with a means of externally controlling and monitoring the sensor systems. All sensor functions available to a local operator of a sensor system that

fully implements the protocol can be commanded via the serial interface. Likewise, all data, status and BIT information that is supplied to the local user can be accessed via the serial interface.

Since the protocol is a command/response protocol the particular information desired from the sensor must be uniquely requested. Requests for information contained in a particular message are made by sending a header only message with the request bit set in the flag field of header word 5. Continuous update of information at a predetermined rate, if supported by the sensor, can be requested by connecting to a given message. A message can be connected by sending a header only message with the connect bit set in the flag field of header word 5.

The numerical identification (NID) for a message is an indication of the intended use of the message. NIDs in the 1000 series, between 1000-and 1999, indicate network control and addressing messages. NIDs in the 2000 series indicate sensor identification information as well as sensor status and BIT. The 3000 series is reserved for sensor command and control messages. Data and information transfer messages, such as those that report target absolute/relative position are included in the 5000 series.

The protocol and message set can also support the pass through of any PLGR specific message. This has been done to support sensors such as the Lightweight Laser Designator Range finder (LLDR) which include an embedded GPS. The PLGR message pass through is generated in the following manner. First the PLGR frame sync is replaced by the Sensor Link Protocol frame sync. The network address field is then added to the standard PLGR message. The NID of the PLGR message contained in header word 3 is then replaced with a PLGR pass through NID. This PLGR pass through NID is calculated by adding the decimal value 10000 to the decimal value of the PLGR NID. A new header checksum is then calculate and the PLGR pass through message is transmitted with the remaining data and header words being sent exactly as they would be if communicating via the PLGR protocol.

A list of supported Sensor Link Protocol messages is included in Table 1. Message data, component indices, modes and mode values are specific to individual sensor types.

Table 1 Existing Message Set

NID	Message Name
0	Universal Reset
1000	Unit Address Request
1001	Unit Address Assign
1002	Unit Sign-on Reject
1003	Group Address Assign
1004	Network Silence
1005	Network Control Transfer
1006	Network Address Table Exchange
2000	Sensor Identification
2001	Perform Built In Test (BIT)
2002	Built In Test (BIT) Status

NID	Message Name
2100	Operational Status
2200	Self Position
3000	Enable Sensor Component
3001	Enable Component Response
3005	Disable Sensor Component
3011	Disable Component Response
3010	Activate Sensor Component
3011	Activate Component Response
3015	De-Activate Sensor Component
3016	De-Activate Component Response
3100	Set Sensor Component Mode
3101	Get Sensor Component Mode
3110	Set Sensor Component Value
3111	Get Sensor Component Value
5000	Absolute Target Position
5001	Relative Target Position
5002	Target Range
5003	Target Angle
7000	Free Text
7100	Bulk Data Transfer

7. Summary

By standardizing on a common interface it becomes possible to interface a variety of sensor system to a diverse set of host computer systems. Standardization based on the RSTA Common Device Protocol provides the maximum interface flexibility specifically tailored for implementation on small man-portable and vehicle mounted sensor systems.

From the system integrator's point of view the Sensor Link Protocol provides a stable sensor interface baseline. The protocol promotes software reuse by providing a common network, data link, and physical interface among a variety of sensor systems. Thus extending the JTA and DII COE concepts to sensor system interfaces.

From the sensor manufacturers point of view the protocol provides a stable interface mechanism, which isolates them from the changing integration responsibilities and requirements associated with interfaces to the tactical internet. Abstracting away mission specific interface issues and reducing associated requirements creep. The protocol, by facilitating software reuse and stabilizing system requirements, also helps reduce system development cost, schedule and risk.

8. References

1 “Reconnaissance Surveillance and Target Acquisition Common Device Protocol - Interface Control Document”, CDP-ICD-200, July 7, 1997.

2 “Department of Defense Joint Technical Architecture Version 1.0”, 22 August 1996.

3 “Department of Defense Joint Technical Architecture Army Version 5.0”, 11 September 1997.