



**(U//FOUO) Intra-Soldier Wireless (ISW)
(U//FOUO) Embedment Guide**

**Version 1.0
September 2018**

DISTRIBUTION STATEMENT D: Distribution authorized to the Department of Defense and U.S. DoD contractors For Official Use Only or for Administrative or Operational Use, 28 September 2018. Other requests for this document shall be referred to U.S. Army Product Manager Ground Soldier, Building 1024, 5966 12th Street, Ft Belvoir, VA 22060

(U) Revision History

Date	Version	Description
September 2018	1.0	Initial document

DRAFT

(U) Table of Contents

1.	(U) Document Purpose and Scope.....	1
2.	(U) Disclaimers.....	1
3.	(U) References.....	1
4.	(U) Definitions.....	1
5.	(U) Acronyms and Abbreviations.....	2
6.	(U) Overview.....	4
7.	(U) Physical Integration.....	4
7.1	(U) Parallel-Attached Module.....	4
7.2	(U) USB-Attached Module.....	8
7.3	(U) Serial-Attached Module.....	10
8.	(U) Application Data Wire Interface Integration.....	14
8.1	(U) Parallel-Attached Module.....	15
8.2	(U) USB-Attached Module.....	16
8.3	(U) Serial-Attached Module.....	17
9.	(U) ISW API Integration.....	18
9.1	(U) Overview.....	18
9.2	(U) General Operation.....	21
9.2.1	(U) API Integration.....	21
9.2.2	(U) API General Rules.....	21
9.2.3	(U) Autonomous Indications.....	22
9.2.4	(U) Node Security.....	22
9.2.5	(U) Node Identity Fields.....	22
9.2.6	(U) Network Association.....	23
9.2.7	(U) Network Join.....	23
9.2.8	(U) Optional Node Management.....	24
9.3	(U) Typical Use Cases.....	24
9.3.1	(U) Start-Up.....	24
9.3.2	(U) Network Association.....	25
9.3.3	(U) Stream Management.....	26
9.3.3.1	(U) Unicast Streams.....	26
9.3.3.2	(U) Multicast Streams.....	26
9.3.3.3	(U) Broadcast Stream.....	26
9.3.4	(U) Firmware Update.....	26
9.3.5	(U) FCC System Certification.....	27
10.	(U) SolNet Integration.....	27
11.	(U) Other Peripheral Interfaces.....	29
11.1	(U) LED.....	29
11.2	(U) PPS.....	29
12.	(U) Antenna Integration.....	29
12.1	(U) Antenna Selection.....	29
12.2	(U) Antenna Placement.....	30
12.2.1	(U) Antenna Location.....	30
12.2.2	(U) Antenna Connection Loss.....	31
12.2.3	(U) Antenna Orientation.....	32
Appendix A.	(U) SolNet Descriptor Examples.....	34

A.1.	(U//FOUO) Weapon Sight to Heads-up-Display	34
A.1.1	(U//FOUO) Weapon Sight Descriptor Set	34
A.1.2	(U//FOUO) Heads-Up-Display Descriptor Set.....	38
Appendix B.	(U//FOUO) FWS to ISW Porting Guidance	40
B.1.	(U) Overview	40
B.2.	(U) Wire Interface.....	40
B.2.1	(U) Word-Aligned API Enforcement.....	40
B.2.2	(U) Peer Indexing Change	40
B.3.	(U) Initialization and Reset.....	42
B.3.1	(U) Autonomous Indications.....	42
B.3.2	(U) Reset Indication and Start-up Sequence.....	42
B.4.	(U) Device Identity	43
B.5.	(U) Network Association	43
B.6.	(U) Idle Scan Frequency	43
B.7.	(U) New Capabilities	43
B.8.	(U) Discovery and Registration (SolNet)	44

DRAFT

(U) List of Figures

Figure 7-1: (U) Parallel-Attached Module Block Diagram	5
Figure 7-2: (U) AL5930 Commander Rev2 Mechanical Dimensions	6
Figure 7-3: (U) AL5930 Commander Rev2 60-Pin Connector Pinout	7
Figure 7-4: (U) USB-Attached Module Block Diagram.....	8
Figure 7-5: (U) AL5934 Combat Rev1 Mechanical Dimensions	9
Figure 7-6: (U) AL5934 Combat Rev2 USB Pinout.....	10
Figure 7-7: (U) Serial-Attached Module Block Diagram	11
Figure 7-8: (U) AL5955 Camouflage Rev2 Mechanical Dimensions	12
Figure 7-9: (U) AL5955 Camouflage Rev2 60-Pin Connector Pinout	13
Figure 7-10: (U) AL5955 Camouflage Rev2 I2C Stake-pin Connections.....	14
Figure 7-11: (U) AL5955 Camouflage Rev2 RS232 Stake-pin Connections	14
Figure 8-1: (U) Parallel Interface Stack.....	15
Figure 8-2: (U) USB Stack	17
Figure 8-3: (U) Serial Stack.....	18
Figure 9-1: (U) API Actions	22
Figure 12-1: (U) Antenna Radiation Plane	30
Figure 12-2: (U) Improper Antenna Placement	31
Figure 12-3: (U) Proper Antenna Placement	31
Figure 12-4: (U) TRDA Chip Antenna.....	32
Figure 12-5: (U) X/Z Radiation Pattern at 3.96 GHz.....	33
Figure 12-6: (U) X/Y Radiation Pattern at 3.96 GHz.....	33

(U) List of Tables

(U) Table 3-1: References.....	1
(U) Table 4-1: Definitions	1
(U) Table 5-1: Acronyms and Abbreviations	2
Table 9-1: (U) API Classes	19
Table 9-2: (U) ISW-Compliant API Elements.....	19
Table 9-2: (U) ISW-Compliant API Elements - Continued.....	20
Table 9-2: (U) ISW-Compliant API Elements - Continued.....	21
Table 10-1: (U) ISW-Compliant SolNet Command/Response Support	28
Table 12-1: (U) WS Label Descriptor.....	34
Table 12-2: (U) WS Video Descriptor Set.....	35
Table 12-3: (U) WS Positioning Descriptor Set	36
Table 12-4: (U) WS Protocol Descriptor Set	37
Table 12-5: (U) WS Service Advertisement	38
Table 12-6: (U) HUD Label Descriptor	38
Table 12-7: (U) HUD Protocol Descriptor Set	39
Table 12-8: (U) HUD Service Advertisement	40
Table 12-9: (U) Peer Indexes	41
Table 12-10: (U) GP_HOST and GP_MAC Register, New	42

1. (U) Document Purpose and Scope

(U//FOUO) This specification describes the general areas to consider when integrating an ISW module. Wherever possible, this document refers to the ISW Protocol Specification for details, or to the vendor to obtain vendor-specific information.

2. (U) Disclaimers

(U) NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

3. (U) References

Table 3-1: (U) References

(U)

Ref #	Title
[R1]	ECMA-368, High Rate Ultra-Wideband PHY and MAC Standard, 3 rd Edition. Also coded as ISO/IEC 26907.
[R2]	ISW Protocol Specification

(U)

4. (U) Definitions

Table 4-1: (U) Definitions

(U)

Term	Definition
Application	Application that uses ISW-Mac/ECMA-368 to transport data
Endpoint	Logical flow of data arriving from an application over a wired interface
Node	The hardware, firmware and software that constitutes an ISW implementation
Peer	Node connected to the local node over the air
Stream	Collection of sub-streams between two peers in a network
Sub-Stream	Logical flow that maps to a logical wired interface endpoint

(U)

5. (U) Acronyms and Abbreviations

Table 5-1: (U) Acronyms and Abbreviations

(U)

Acronym	Abbreviation
AFE	Analog Front End
AL	Alereon
ASCII	American Standard Code for Information Interchange
ATECC	Microchip ATECC608A Security Processor
API	Application Programming Interface
BBP	Baseband Processor
BPF	Bandpass Filter
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance
CERDEC	Communications, Electronics Research, Development & Engineering Center
CF+	Compact Flash Plus
DAGR	Defense Advanced GPS Receiver
ECMA-369	European Computer Manufacturers Association - 369
ENVG-III	Enhanced Night Vision Goggle-III
ENVG-B	Enhanced Night Vision Goggle-Binocular
EUI-48	End User Identifier, 48 bits
FCC	Federal Communications Commission
FPGA	Field Programmable Gate Array
FOUO	For Official Use Only
FWS-CS	Family of Weapon Sights-Crew Served
FWS-I	Family of Weapon Sight-Individual
GPIO	General Purpose Input/Output
GSSIP	GPS Standard Serial Interface Protocol
I2C	Inter-Integrated Circuit (pronounced "I-squared-C")

(U)

(U)

Acronym	Abbreviation
IMU	Inertial Measurement Unit
I/O	Input/Output
ISW	Intra-Soldier Wireless
LED	Light Emitting Diode
LSB	Least Significant Bits
MAC	Media Access Control
MCU	Microcontroller Unit
MSB	Most Significant Bits
NIST	National Institute of Standards and Technology
NVRAM	Non-volatile RAM
PGDP	Pretty Good Device Protocol
PHY	Physical Layer
PPS	Pulse Per Second
QoS	Quality of Service
SBP	Smart Battery Pack
SPI	Serial Peripheral Interface
SolNet	Intra-Solider Wireless Networking
SWIM	Simple Wire Interface Mux/De-mux
TWS	Thermal Weapon Sight
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
UWB	Ultra-Wideband
WS	Weapon Sight

(U)

6. (U) Overview

(U) To embed an ISW module, the integrator must:

- (U//FOUO) Physically embed the module
- (U//FOUO) Logically integrate the appropriate application data wire interface used for sending data to and receiving data from the module
- (U//FOUO) Support the required subset of ISW API functions to control the module
- (U//FOUO) Support the required subset of SolNet protocol to control discovery and service registration
- (U//FOUO) Integrate the antenna appropriately
- (U//FOUO) Integrate other peripheral interfaces

(U) The remaining sections of the document describe these tasks relative to three ISW modules:

- (U//FOUO) Parallel-attached module
- (U//FOUO) USB-attached module
- (U//FOUO) Serial-attached module

(U//FOUO) For ISW compliance, the integrator must support a required subset of ISW API commands and a required subset of SolNet protocol commands. ISW API compliance is described in the ISW API Integration section. ISW SolNet compliance is described in the SolNet Integration section. Appendix A shows an example of creating a compliant system. Appendix B gives guidance for porting from legacy FWS systems to ISW.

7. (U) Physical Integration

(U//FOUO) On delivery from the vendor, each module has been calibrated and verified operational. How the module is integrated depends on the module type and the nature of the system that uses the module. The module includes conformal coating over the security elements. The integrator must take care to avoid damaging the coating when embedding the module.

(U//FOUO) For each module, a GPIO line is brought out to a resistor to support PPS operation. Contact the module vendor for the location of the resistor. Refer to the Other Peripheral Interfaces section for PPS operation.

(U//FOUO) Each module includes an LED. The LED visually illustrates certain module operational states. The integrator may remove the LED or bring the LED out from the board. Refer to the Other Peripheral Interfaces section for LED operation.

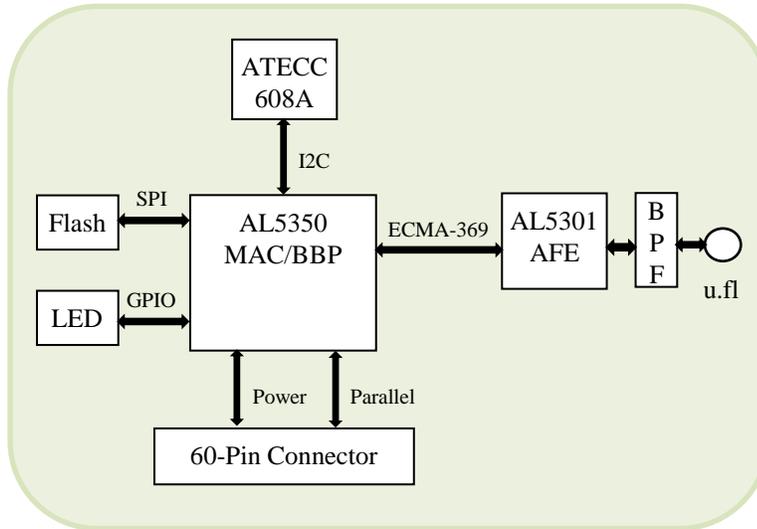
7.1 (U) Parallel-Attached Module

(U//FOUO) The parallel-attached module exposes a 60-pin connector (Hirose DF40C). This connector brings out 3.3V power, parallel interface lines, the power-on-reset pin, and calibration pins. The host system must connect up the power, parallel interface lines, and power-on-reset pins only.

(U//FOUO) The following diagram shows the basic blocks of the module.

Figure 7-1: (U) Parallel-Attached Module Block Diagram

(U//FOUO)



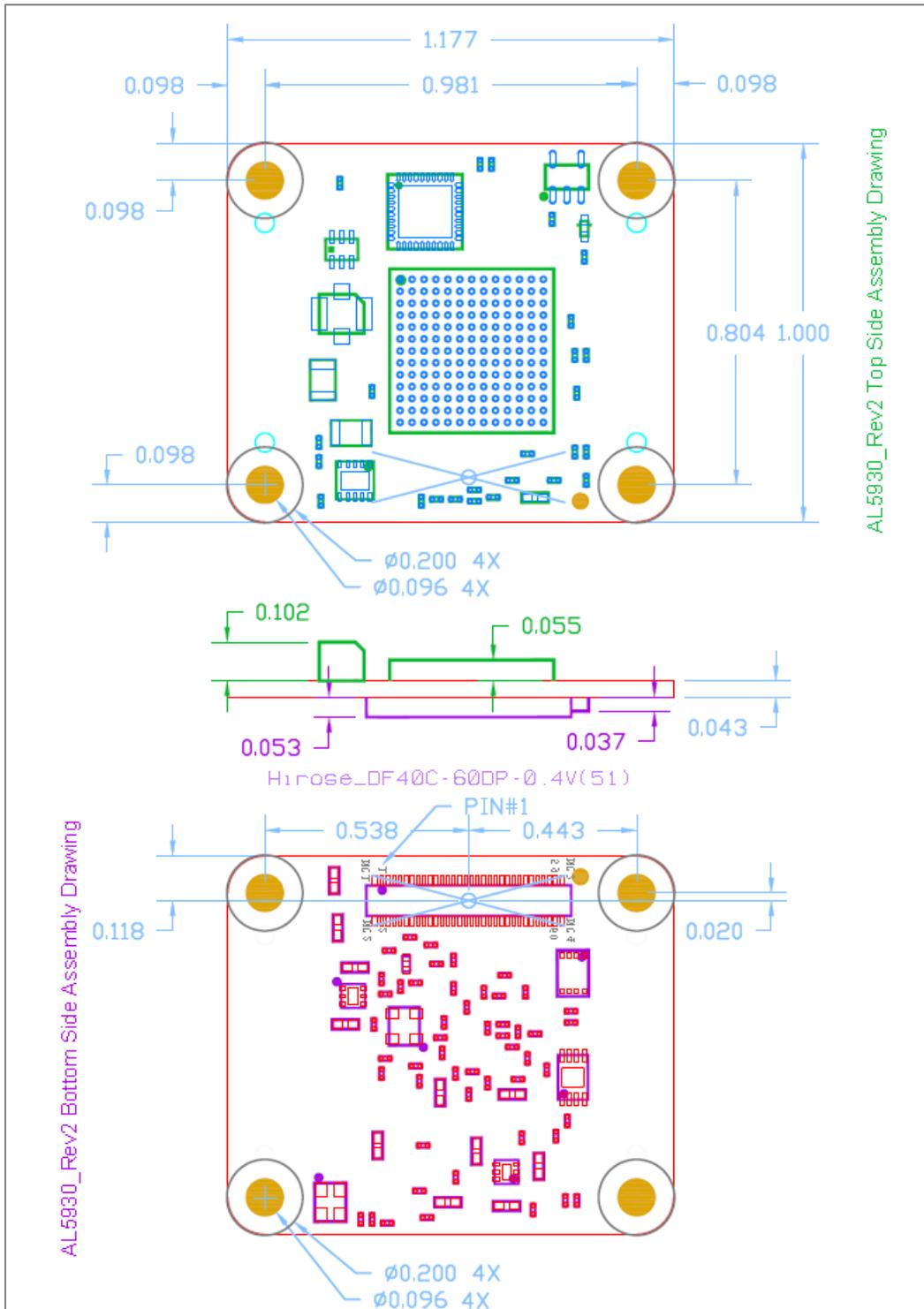
(U//FOUO)

(U//FOUO) This module supports a u.fl connector for an external antenna. Refer to the Antenna Integration section for details.

(U//FOUO) The following figure shows the mechanical layout, dimensions, and pinout.

Figure 7-2: (U) AL5930 Commander Rev2 Mechanical Dimensions

(U//FOUO)



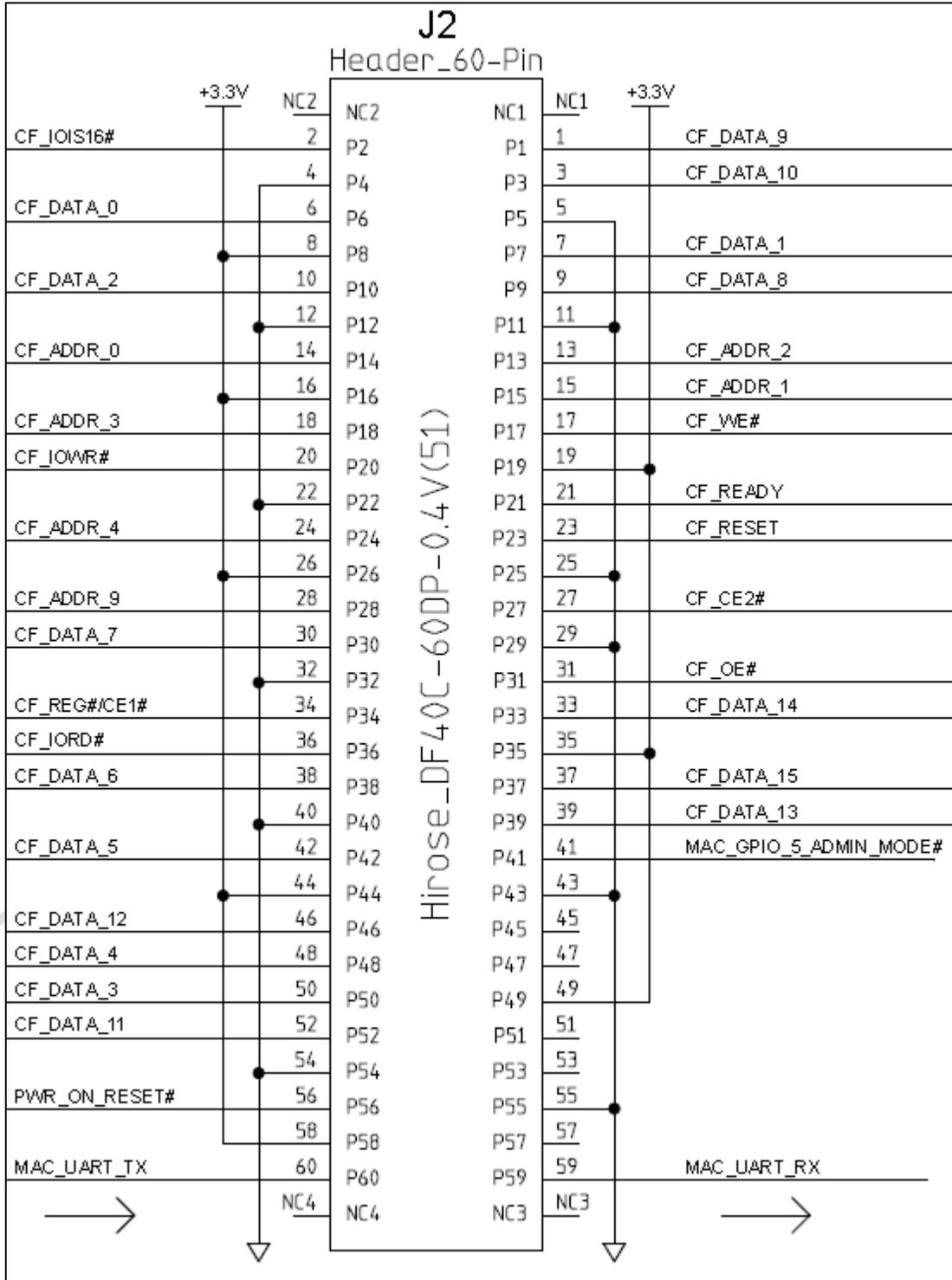
AL5930_Rev2 Top Side Assembly Drawing

AL5930_Rev2 Bottom Side Assembly Drawing

(U//FOUO)

Figure 7-3: (U) AL5930 Commander Rev2 60-Pin Connector Pinout

(U//FOUO)

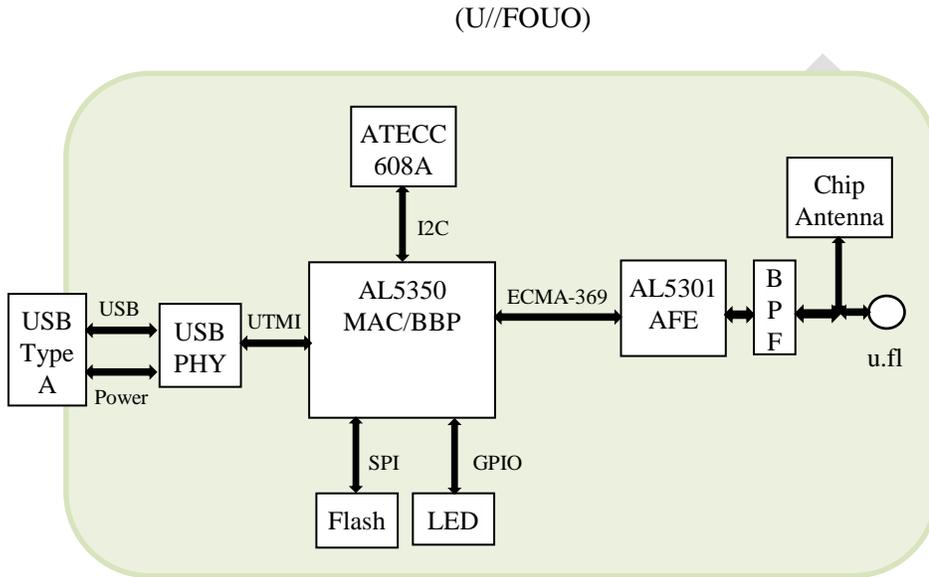


(U//FOUO)

7.2 (U) USB-Attached Module

(U//FOUO) The USB-attached module exposes a USB Type A male connector with D+, D-, 5V, and ground. The connector may be removed for integration into an embedded system. The USB ID pin is pulled high, which causes the module to connect to a host system as a USB device. The following diagram shows the basic blocks of the module.

Figure 7-4: (U) USB-Attached Module Block Diagram



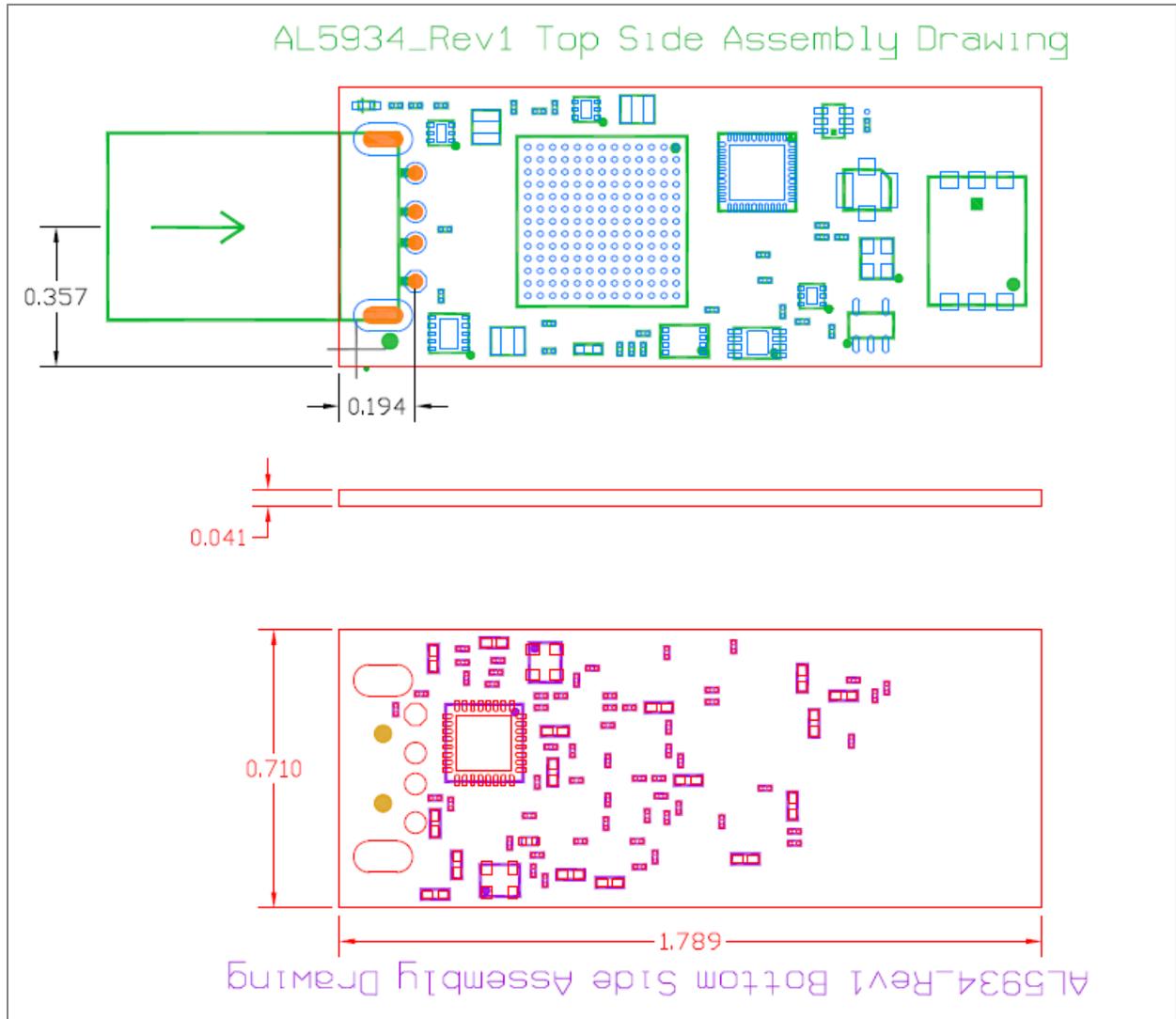
(U//FOUO)

(U//FOUO) Each module supports an on-board chip antenna and a u.fl connector for an external antenna. Refer to the Antenna Integration section for details.

(U) The following figure shows the mechanical layout, dimensions, and pinout.

Figure 7-5: (U) AL5934 Combat Rev1 Mechanical Dimensions

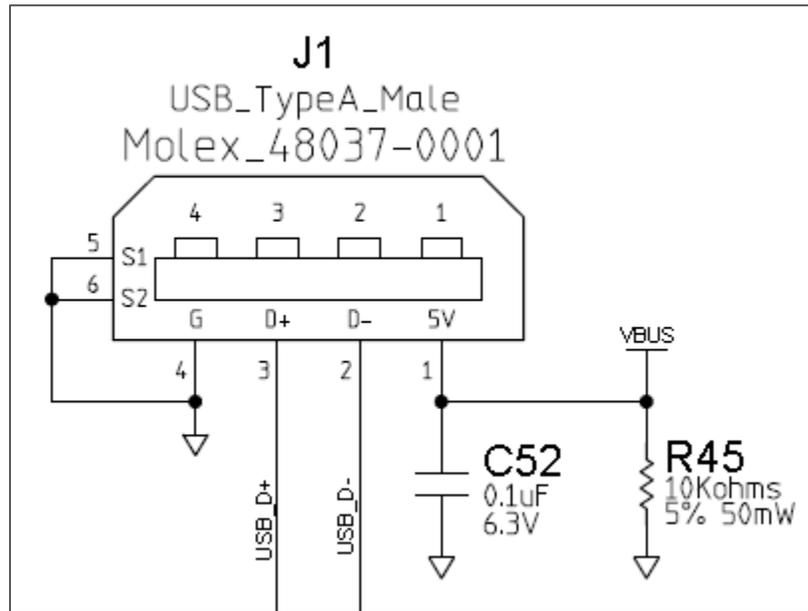
(U//FOUO)



(U//FOUO)

Figure 7-6: (U) AL5934 Combat Rev2 USB Pinout

(FOUO)



(U//FOUO)

7.3 (U) Serial-Attached Module

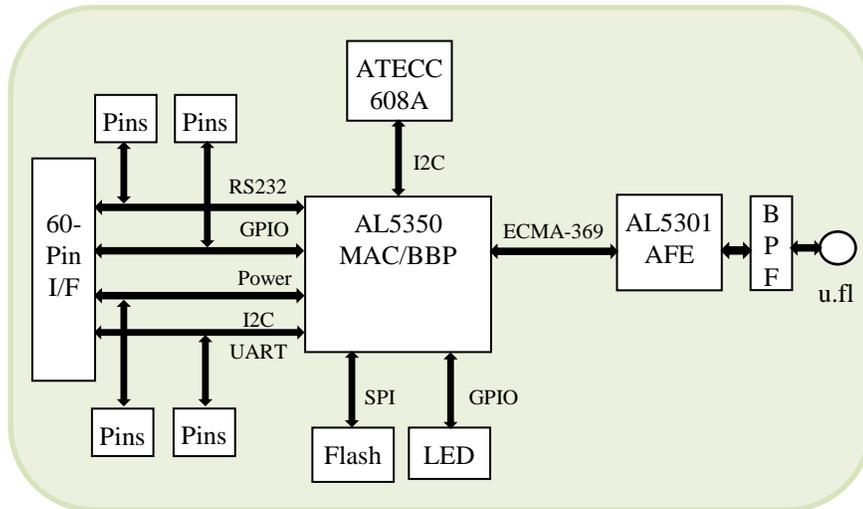
(U//FOUO) The serial-attached module exposes a 60-pin connector (Hirose DF40C). This connector brings out power, I2C/UART lines, RS232 lines, a GPIO line, the power-on-reset pin, and calibration pins. In addition to the connector, the module exposes individual through holes for 3.3V power, I2C/UART lines, RS232 lines, and a GPIO line. The integrator may use the 60-pin connector to integrate to a mother board or may integrate directly via the through-hole interfaces.

(U//FOUO) Because the I2C and UART lines are shared, the integrator may use these lines as I2C or UART, but not both at the same time. If the RS232 interface is in use, then the I2C/UART lines are not available.

(U) The following diagram shows the basic blocks of the module.

Figure 7-7: (U) Serial-Attached Module Block Diagram

(U//FOUO)



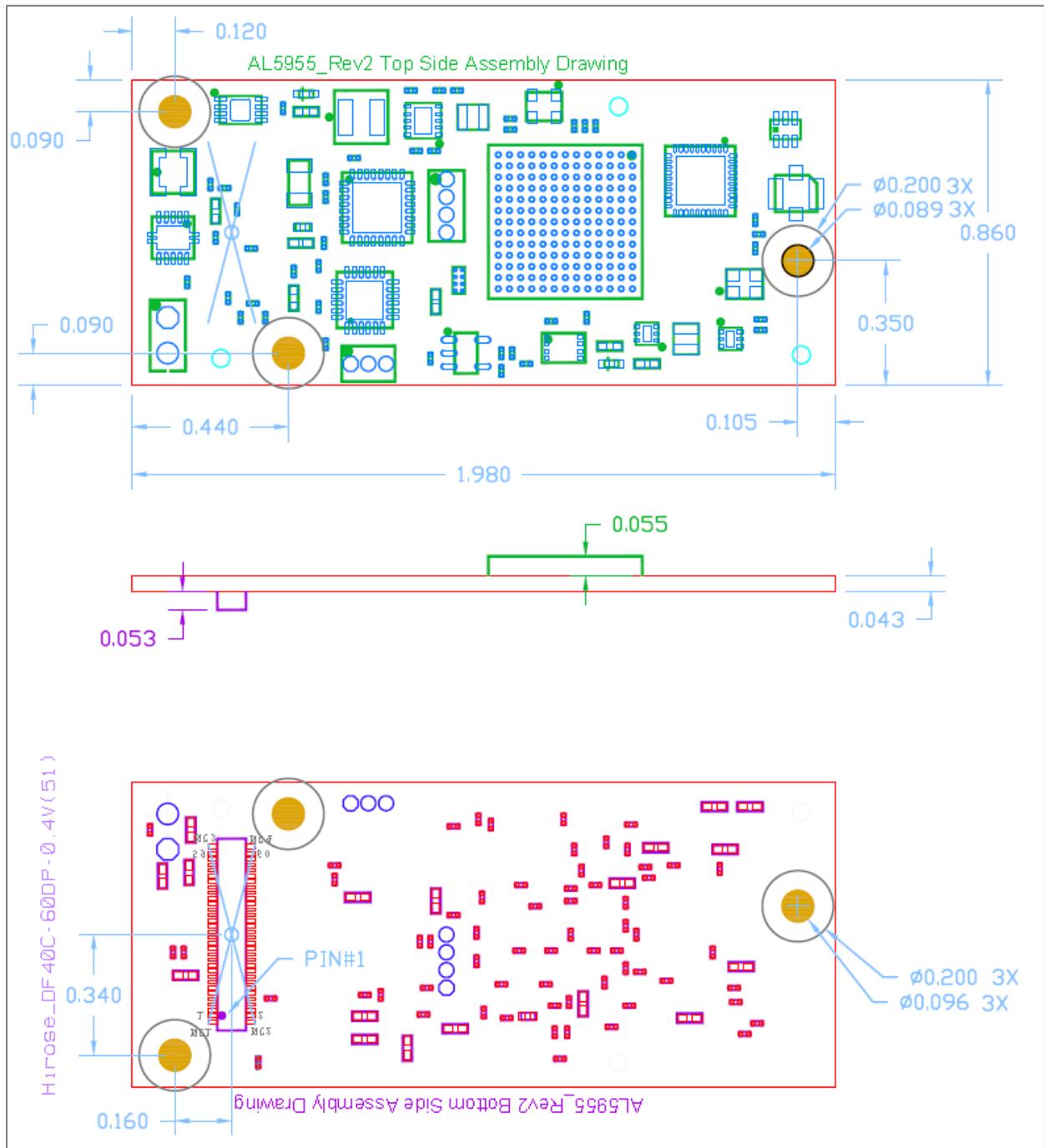
(U//FOUO)

(U) This module supports a u.fl connector for an external antenna. Refer to the Antenna Integration section for details.

(U) The following figure shows the mechanical layout, dimensions, and pinout.

Figure 7-8: (U) AL5955 Camouflage Rev2 Mechanical Dimensions

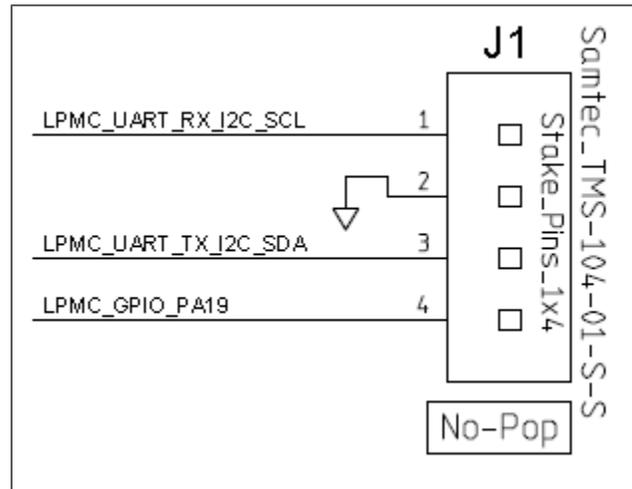
(U//FOUO)



(U//FOUO)

Figure 7-10: (U) AL5955 Camouflage Rev2 I2C Stake-pin Connections

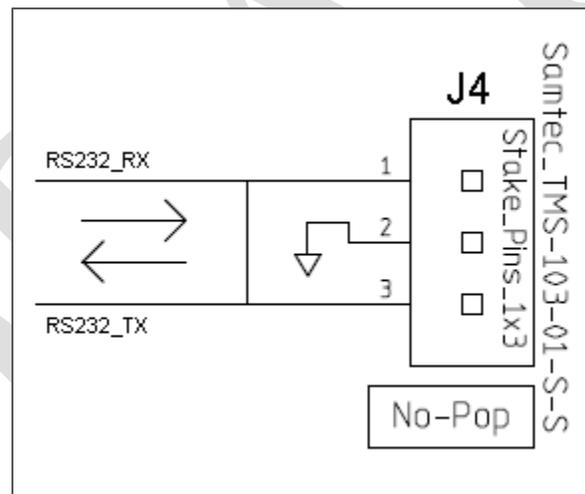
(U//FOUO)



(U//FOUO)

Figure 7-11: (U) AL5955 Camouflage Rev2 RS232 Stake-pin Connections

(U//FOUO)



(U//FOUO)

8. (U) Application Data Wire Interface Integration

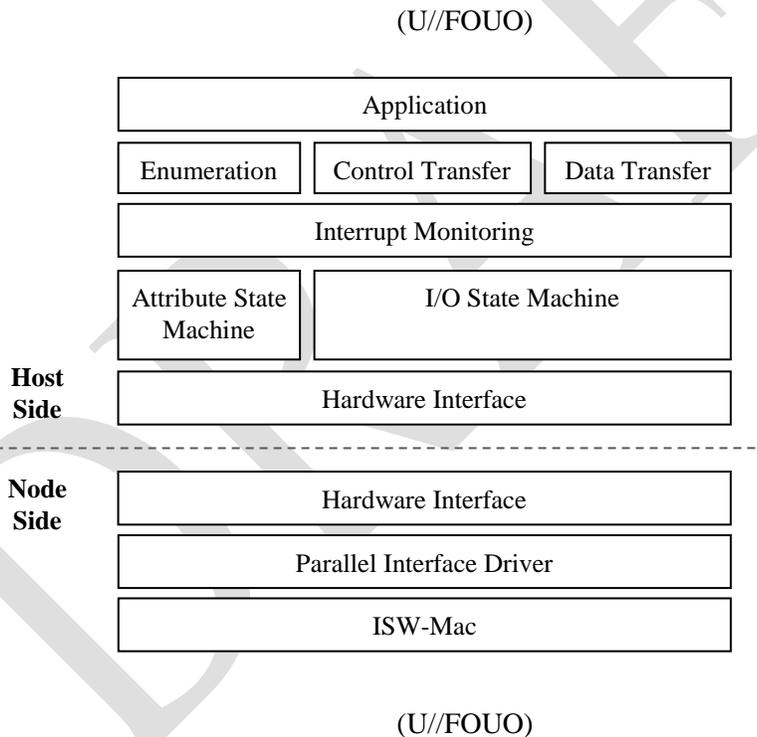
(U//FOUO) Every ISW module supports one application data wire interface. This interface supports application data exchange with the ISW module. Application data is sent to or received from a peer. Sent data is automatically encrypted by the ISW node, placed on air, and decrypted by the receiving node before sending to the receiving application over the wire interface. Each packet references a particular peer via a peer index, and a particular endpoint. The application may send application data over any or all of four endpoints numbered 0 – 3. Packets over endpoint 0 must be encoded with a SolNet header. Packets over endpoints 1 – 3 may optionally be encoded with SolNet headers at the integrator's discretion.

8.1 (U) Parallel-Attached Module

(U//FOUO) The parallel-attached module attaches through a memory-mapped interface based on a subset of the CompactFlash Plus (CF+) specification. This interface may be integrated to an application system via an FPGA or a MCU that supports programmable I/O state machines or a hardware-based memory mapped interface. Contact the module vendor for documentation on the pinout, registers, timing, and states of the interface.

(U//FOUO) The parallel-attached Node requires enumeration before use. The enumeration begins when the Node alerts the Host that the Node is ready for enumeration. The Host performs enumeration, which consists solely of writing a set of values to the attribute registers. Following enumeration, the Host may write and read control packets or application data packets using Node interrupts to determine when reading/writing of packets is allowed. When writing packets, the Host sets the peer index and endpoint of the recipient. The peer index determines if the packet is an application data packet or a control packet. Refer to the module vendor documentation for how to set this information. When reading packets, the Host determines the peer index and endpoint of the sender, where peer index may identify a packet from a peer or a control packet from the Node. Refer to the module vendor documentation for how to determine this information. The following figure describes the Host side of the parallel interface stack.

Figure 8-1: (U) Parallel Interface Stack



(U) The stack elements are:

- (U//FOUO) Host Hardware Interface – The parallel interface pins physically connecting to the corresponding pins of the Node.
- (U//FOUO) Attribute State Machine – The state machine that controls state and timing for accessing the Node's attribute registers. The attribute registers are accessed during interface enumeration. The real-time nature of the interface requires implementation in hardware, such as an FPGA, programmable gates, or MCU memory-mapped interface.

- (U//FOUO) I/O State Machine – The state machine that controls state and timing for accessing the Node's I/O registers. The I/O registers are accessed for sending and receiving application data and application control commands. The real-time nature of the interface requires implementation in hardware, such as an FPGA, programmable gates, or MCU memory-mapped interface.
- (U//FOUO) Interrupt Monitoring – The logic that monitors the interrupt line that indicates when a packet has arrived for reading and when a new packet may be written. This can be done in software.
- (U//FOUO) Enumeration – The logic that controls the enumeration process. This can be done in software.
- (U//FOUO) Control Transfer – The logic that sends ISW API command packets to the Node and receives ISW API responses and indications from the Node. This block sets or decodes the peer index and endpoint. This can be done in software.
- (U//FOUO) Data Transfer – The logic that sends application data packets to the Node and receives application data packets from the Node. This block sets or decodes the peer index and endpoint. This can be done in software.

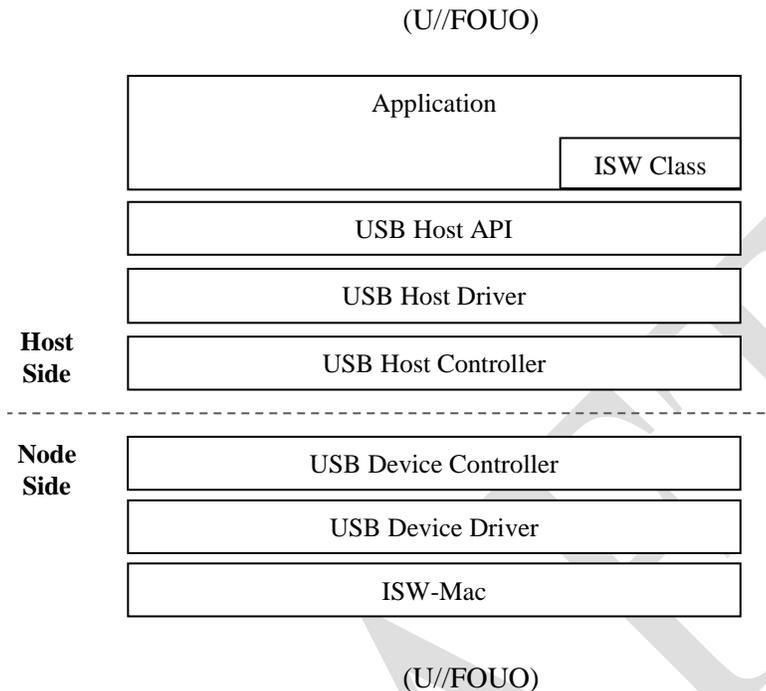
8.2 (U) USB-Attached Module

(U//FOUO) the USB-attached module attaches through a standard USB bus. The Host serves as a USB Host while the Node operates as a Vendor USB Device. This interface may be integrated to an application system via an FPGA or a MCU that implements a USB Host. Contact the module vendor for documentation on the descriptors and detailed operation of the USB Device.

(U//FOUO) The USB Device requires enumeration before use. The Node responds to standard USB enumeration, presenting itself as a Vendor-defined USB Device. The device descriptors describe two endpoints: a Bulk IN and a Bulk OUT. Following enumeration, the Host may write control packets or application data packets over the OUT endpoint and read them over the IN endpoint. When writing packets, the Host prepends a SWIM header to indicate the peer index and endpoint of the recipient. The peer index determines if the packet is an application data packet or a control packet. When reading packets, the Host decodes the SWIM header to determine the sending peer index and endpoint, where peer index may identify a packet from a peer or a control packet from the Node. Refer to the ISW Protocol Specification for the format of the SWIM header.

(U) The following figure describes the Host side of the USB stack.

Figure 8-2: (U) USB Stack



(U) The stack elements are:

- (U//FOUO) USB Host Controller – The Host controller hardware and Host controller driver.
- (U//FOUO) USB Host Driver – The USB Host driver that performs enumeration and manages data transfers.
- (U//FOUO) USB Host API – The API to the USB Host Driver as presented by the particular USB Host stack.
- (U//FOUO) ISW Class – ISW Class specification/driver. This block sets or decodes the peer index and endpoint (SWIM header).

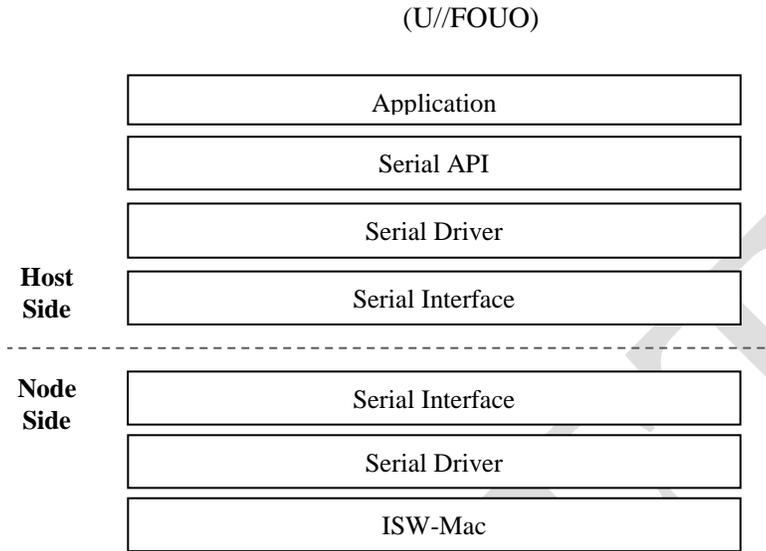
8.3 (U) Serial-Attached Module

(U//FOUO) The Serial-attached module attaches through a standard UART or I2C interface. In the case of the I2C interface, the Host is master. Contact the module vendor for documentation on the serial settings.

(U//FOUO) The Host may send or receive control packets or application data packets over the serial interface. When sending packets, the Host prepends a SWIM header to indicate the peer index and endpoint of the recipient. The peer index determines if the packet is an application data packet or a control packet. When receiving packets, the Host decodes the SWIM header to determine the sending peer index and endpoint, where peer index may identify a packet from a peer or a control packet from the Node. Refer to the ISW Protocol Specification for the format of the SWIM header.

(U) The following figure describes the Host side of the serial interface.

Figure 8-3: (U) Serial Stack



(U//FOUO)

(U) The stack elements are:

- (U//FOUO) Serial Interface – The hardware interface (UART or I2C master).
- (U//FOUO) Serial Driver – The driver that sends or receives characters or packets.
- (U//FOUO) Serial API – The API to the Serial Driver. This block sets or decodes the peer index and endpoint (SWIM header).

9. (U) ISW API Integration

9.1 (U) Overview

(U//FOUO) The ISW node implements a low-touch control model where the application need not control the node at real-time rates. The ISW API allows the application to perform certain infrequent operations, set values that are used by real-time processes in the node, and monitor status of the node and connected nodes. The API includes commands, responses to commands, and autonomous indications.

(U) The following table lists the classes of API commands.

Table 9-1: (U) API Classes

(U//FOUO)

Class	Description
Firmware Update	The application may query the firmware version(s) and upgrade firmware.
Identity	The application may get or set identity fields.
Security	The application may perform operations related to a secure module.
Association	The application may initiate the joining of a network.
Streams	The application may monitor the status of streams (connected peers), establish multicast groups, and set QoS parameters for streams.
Link	The application may set parameters for system level link operation, including scan characteristics and service intervals.
Metrics	The application may get wired or wireless statistics.
System	The application may set system modes and reset the node.

(U//FOUO)

(U//FOUO) The ISW Protocol Specification describes the APIs in detail. The APIs apply to all module types.

(U//FOUO) For ISW compliance, the following table shows the commands, responses, and indications that are required to support ISW-compliant operation, versus those that are optional. Those marked as 'conditional' indicate elements that may be required depending on the application implementation.

Table 9-2: (U) ISW-Compliant API Elements

(U//FOUO)

	Compliance	Note
Firmware Update		
Get Image Version	<i>Required</i>	
Load Image, Read Image, Run Image	<i>Optional</i>	Supports over-the-wire firmware update
Identity		
Set Device Type, Get Device Type	<i>Required</i>	
Set Coordinator, Get Coordinator	<i>Required</i>	
Set MAC Address, Get MAC Address	<i>Optional</i>	MAC address should not be changed typically
Security		
Authenticate Role	<i>Required</i>	

(U//FOUO)

Table 9-3: (U) ISW-Compliant API Elements - Continued

(U//FOUO)

	Compliance	Note
Start Crypto Session	<i>Required</i>	
Associations		
Start Association	<i>Required</i>	
Clear Association	<i>Required</i>	
Association Indication	<i>Required</i>	
Streams		
Set Stream Spec, Get Stream Spec	<i>Optional</i>	Supports quality of service for the network
Get Streams Status	<i>Optional</i>	Superseded by Get Extended Streams Status
Get Extended Streams Status	<i>Conditional</i>	Use this if polling for status; otherwise, use Streams Status Indication
Start Multicast Group, Join Multicast Group, Leave Multicast Group	<i>Required</i>	
Modify Multicast Group	<i>Optional</i>	Supports setting PHY rate of multicast transmits
Get Multicast Groups Status	<i>Conditional</i>	Use this if polling for status; otherwise, use Multicast Groups Status Indication
Get Peer Firmware Version	<i>Optional</i>	Supports getting firmware version from peers
Connection Indication	<i>Optional</i>	Info contained by Get Extended Streams Status response and Streams Status Indication
Streams Status Indication	<i>Conditional</i>	Use this for autonomous status; otherwise, use Get Extended Streams Status
Multicast Activity Indication	<i>Optional</i>	Info contained by Get Multicast Groups Status response and Multicast Groups Status Indication
Multicast Groups Status Indication	<i>Conditional</i>	Use this for autonomous status; otherwise, use Get Multicast Groups Status
Link		
Set Scan Duty Cycle, Get Scan Duty Cycle	<i>Optional</i>	Supports power management for non-coordinator nodes
Set Idle Scan Frequency, Get Idle Scan Frequency	<i>Required</i>	
Set Maximum Service Interval	<i>Optional</i>	Supports quality of service
Reset Service Interval	<i>Optional</i>	Supports quality of service

(U//FOUO)

Table 9-4: (U) ISW-Compliant API Elements - Continued

(U//FOUO)

Metrics		
Get Wired Metrics	<i>Optional</i>	Supports performance reporting
Get Wireless Metrics	<i>Optional</i>	Supports performance reporting
System		
Reset	<i>Required</i>	
Set Indications, Get Indications	<i>Optional</i>	Supports suppression of indications
Reset Radio	<i>Optional</i>	Supports recovery from power-saving mode
Idle Radio	<i>Optional</i>	Supports initiation of power-saving mode
Reset Indication	<i>Required</i>	

(U//FOUO)

(U//FOUO) The General Operation section describes the basic API concepts and operations. The Typical Use Cases section describes how the API should be used for typical operation, including sequence.

9.2 (U) General Operation

9.2.1 (U) API Integration

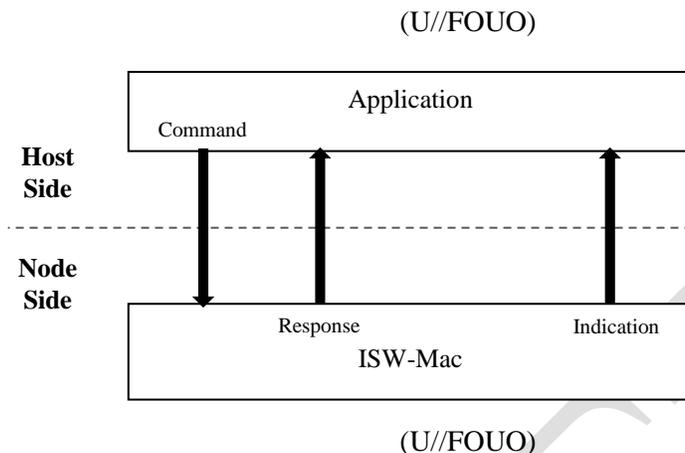
(U//FOUO) When integrating the API to the wired interface, the integrator should verify operation using the *Get Firmware Version* command. This command returns an ASCII string that produces a readable firmware version. This command can be used to verify endianness of the command and response, has a low likelihood of being mistaken for a different command, and has no lasting effect on the node. Endianness rules are defined in the ISW Protocol Specification.

9.2.2 (U) API General Rules

(U//FOUO) The ISW API views the interface in terms of commands, responses, and indications.

(U) The general picture is shown in the following figure.

Figure 9-1: (U) API Actions



(U//FOUO) Every command results in a quick response (usually within milliseconds, but perhaps up to one second). Some commands initiate long-running operations, such as the *Start Association* command. The node returns an immediate response to such commands, and then returns an association indication when the operation completes (e.g. *Association Indication*).

(U//FOUO) Every response and indication includes a result code. Any value other than zero indicates an error.

(U//FOUO) The ISW API supports up to five outstanding commands at one time. This restriction prevents the application from overwhelming the node's memory with commands.

9.2.3 (U) Autonomous Indications

(U//FOUO) By default, the ISW node sends autonomous indications to the application. At power-up, the node issues a *Reset Indication* packet to the application. It then issues a once-per-second status of all streams as a *Streams Status Indication* version 2 packet. On first power-up, the *Streams Status Indication* will show no peer nodes. After a node is part of a network, it issues *Connection Indication* packet each time a node connects to or disconnects from the network. When the node joins a network for the first time, it issues an *Association Indication* packet to the application. The application may disable or enable which indications it wishes to see at any time using the *Set Indications* packet.

9.2.4 (U) Node Security

(U//FOUO) To support node security, the ISW node requires the application to provide a password and a role before executing security-related commands. The login command is *Authenticate Role*.

(U//FOUO) To initiate security functions, the ISW node requires the application to start cryptographic operation. The command to start cryptographic operation is *Crypto Session Start*.

(U//FOUO) These two commands must be run as soon as possible after receipt of any *Reset Indication*.

9.2.5 (U) Node Identity Fields

(U//FOUO) The ISW node contains three forms of identity:

- (U//FOUO) Device type – A general tag that allows the integrator to define different types of nodes.

- (U//FOUO) EUI-48 address – The MAC address of the node. Note: This is set at the factory, so overwriting the address is not recommended.
- (U//FOUO) Coordinator Role – A specification of the node's ability to act as a coordinator.

(U//FOUO) By default, the EUI-48 address is set to a unique value, device type is set to an unspecified value, and the coordinator role is not set. On first power-up, the application should get the device type using a *Get Device Type* packet. If the device type is not the desired value, the application should set it to the desired value using the *Set Device Type* packet. The new device type is automatically stored in non-volatile memory.

(U//FOUO) Every device type exhibits a default behavior in terms of whether or not it can serve as a coordinator and what its coordinator precedence is. If the default behavior of the device type is not acceptable, the application can override the default behavior using the *Set Coordinator* packet. The application may optionally specify if this override should be stored in non-volatile memory. Before setting the override, the application can determine if the override has already been written by sending a *Get Coordinator* packet.

(U//FOUO) To prevent flash fatigue, the application should get the device type and coordinator role and only write them if the stored values are not the desired values.

9.2.6 (U) Network Association

(U//FOUO) By default, an ISW node is not part of a network. On first power-up, it generates a random Network ID and Network Password. The Network ID is 32 bits, so the odds of adjacent nodes generating the same Network ID is infinitesimally small. The application may cause a node to join an existing network by issuing to the node a *Start Association* packet with a type field of 'search'. At the same time, the application on a node that is already part of the target network must issue a *Start Association* packet with a type field of 'invite'. The searching node scans the channel map to find the inviting node. Upon finding the one another, the two nodes perform a process whereby the searching node joins the network of the inviting node. The searching node adopts the Network ID of the new network, and is given the Network Password over a secure tunnel. When association completes, each node issues an *Association Indication* packet to its application. The result code of the packet indicates success, failure, or timeout. The node remains joined to the network after association.

(U) To remove a node from a network, the application can either:

- (U//FOUO) Issue a *Start Association* packet with a type field of 'search' to join it to a new network. The initiation of this operation immediately clears the existing Network ID. If the node does not find another network with which to associate before the association timeout, it generates a new random Network ID.
- (U//FOUO) Issue a *Clear Association* packet to immediately clear the existing Network ID. The node automatically generates a new random Network ID.

9.2.7 (U) Network Join

(U//FOUO) Once a node is associated with a network, after each power-up, the node automatically searches for an already established network advertising its Network ID. The application need take no action to cause this operation to occur. Upon finding a network advertising its Network ID, the node automatically joins the found network. If the node does not find an already established network within one sweep of the channel map and is coordinator-capable, it forms a network on a clear channel with itself as the only member. Other nodes may join the new network later. If the node is not coordinator-capable, it continues scanning for a network to join. While searching for an already established network, the node ignores networks that do not advertise its Network ID.

(U//FOUO) Members of a network are called *peers*. The ISW node reports peer connections to the application as *Connection Indications* and once-per-second *Streams Status Indications*. The application processes these messages to determine when nodes join and leave the network. Every time the ISW node detects a new node, it assigns a local peer index to the node. The peer index is simply a temporary shorthand address that allows the application to target the peer for sent data or determine which peer sent received data. The peer index is assigned first-come, first-served after joining a network. As a result, the peer index for a particular peer may change between power cycles or leave/join cycles. The peer index is only meaningful to the local node.

9.2.8 (U) Optional Node Management

(U//FOUO) The ISW API provides a number of optional commands to manage, configure, or monitor the node. These include:

- (U//FOUO) Setting the scan duty cycle. When a node is not coordinator-capable, it will scan indefinitely for a network to join. This can consume power if no network is available. The *Set Scan Duty Cycle* packet causes the searching node perform a duty cycle between searching and sleeping.
- (U//FOUO) Setting the idle scan frequency. When a node is acting as the coordinator, it periodically scans background channels to a) maintain a list of clear channels to which to escape in the event of sudden jamming and b) search for members of its network on other channels so that the two networks might merge. The background scan consumes power. The *Set Idle Scan Frequency* packet allows the application to balance power consumption and background scan aggressiveness.
- (U//FOUO) Setting per-stream and per-substream quality of service (QoS) behavior. By default, streams and substreams are processed round robin. In the event of local congestion, nodes do not discard packets already received. The *Set Stream Spec* command allows the application to limit the service latency per stream, where a stream is a connection to a particular peer. The same command also allows the application to specify the service priority of sub-streams within a stream, and whether or not to discard packets in the event of local congestion.
- (U//FOUO) Managing multicast groups. The ISW API provides a set of commands and indications that allow the application to start a multicast group, join a multicast group, leave a multicast group, and monitor for multicast groups.
- (U//FOUO) Managing system behavior. The ISW API provides a set of system commands that allow constraining of the service interval for the entire network, placing the radio into an idle state, resuming the radio from the idle state, and hard resetting the node.
- (U//FOUO) Updating firmware. The ISW API provides a set of commands that allow the application to query and update firmware on the node.
- (U//FOUO) Performing security operations. The ISW API provides security commands associated with security certified products.

9.3 (U) Typical Use Cases

9.3.1 (U) Start-Up

(U//FOUO) Assuming that a set of ISW nodes have already been associated, the typical start-up sequence is as follows:

- (U//FOUO) The Host initializes the wire interface (enumeration).

- (U//FOUO) By default the ISW node sends autonomous indications to the application. If the application does not wish to receive certain indications, it issues a *Set Indications* packet. At minimum, the application should keep the following indications enabled: *Wire Bus Enable*, *Reset Indication*. The primary reason for disabling any indication is the turning off of *Connection Indication* in lieu of solely using the *Streams Status Indication*.
- (U//FOUO) After initializing the wireless interface, the application waits for a *Reset Indication* with a result code of success. This indicates that the module has initialized and has passed all of the security and integrity self-tests.
- (U//FOUO) As soon as possible, the application logs in as the crypto officer by issuing an *Authenticate Role* packet with the appropriate password and the crypto officer role. Contact the vendor for the password.
- (U//FOUO) As soon as possible after logging in, the application start cryptographic operation by issuing a *Crypto Session Start* command. This notifies the module that it may begin using cryptographic functions, including key agreement and encryption. The module will not function until this command is entered. This command may not be entered until the application has logged in.
- (U//FOUO) The application reads the device type (*Get Device Type*) and coordinator role (*Get Coordinator*). If the values are not appropriate, the application sets the device type (*Set Device Type*) and/or coordinator role (*Set Coordinator*).
- (U//FOUO) Even though *Streams Status Indication* is enabled by default, it only outputs once per second. On initial power-up, the application may issue a *Get Extended Streams Status* packet to immediately learn the state of the system. If the status shows at least one peer and it is *connected*, at least one stream is present. The application may begin sending data to that peer using the assigned peer index to target the peer. If the status shows that no peers are connected, no peer is connected. The application must wait for an autonomous *Connection Indication* from the node or a *Streams Status Indication* showing at least one connected peer. Upon receiving that indication, the system may begin sending data to that peer using the appropriate peer index. *Note: Integrators have found the once-per-second Streams Status Indication to be the most reliable and simplest to process mechanism, and often use this instead of Connection Indication to track current status of all links.*
- (U//FOUO) The application continues monitor the streams status. As peers join, the application uses the assigned peer index to send data to that peer or determine that data has arrived from that peer. As peers leave, the application stops sending data to the departed peer. When a peer departs, the ISW node automatically flushes any data queued for the peer.

9.3.2 (U) Network Association

(U//FOUO) Assuming two systems, A and B, which have different Network IDs, and System B wishes to join System A's network:

- (U//FOUO) System A application issues *Start Association* packet with type = invite and association timeout of T. Node returns response.
- (U//FOUO) System B application issues *Start Association* packet with type = search and association timeout of T. Node returns response.
- (U//FOUO) Note: The order of placing systems into association mode does not matter as long as the second is placed into association mode at least 6 – 8 seconds before the association timeout of the first system.

- (U//FOUO) Upon successful association, both nodes return *Association Indication* with result code of success to respective applications.
- (U//FOUO) If a node does not detect one another before association timeout, it returns *Association Indication* with result code of timeout to application.
- (U//FOUO) If a node begins the association sequence with another node, but fails due to poor RF conditions or a security violation, the node returns *Association Indication* with result code of general failure to application.

(U//FOUO) For robust association, the lowest allowed timeout is 30 seconds. Under typical user scenarios, associations of longer than 60 seconds are excessive.

9.3.3 (U) Stream Management

9.3.3.1 (U) Unicast Streams

(U//FOUO) After initializing the ISW node, secure unicast connections form automatically amongst associated nodes. The active streams are reported by the Streams Status Indication. The application tracks which peer index is used by which peer node. The Streams Status Indication includes the EUI-48 address and device type to help the application uniquely identify the peer.

9.3.3.2 (U) Multicast Streams

(U//FOUO) Multicast streams are not automatically established. They must be started/stopped and joined/left by the application. The application may start one multicast group and join one multicast group (different groups). For each of the two groups, the application may send or receive data on the group.

(U//FOUO) The application starts a multicast group using the *Start Multicast Group* command. This command assigns a multicast address that is advertised to all peers in a *Multicast Activity Indication* with a status of "found". The *Multicast Group Status* shows available groups to peers. If no peers are connected, the application reports a status of paused. Regardless, the application may begin sending multicast data using the multicast peer index returned by the start command. If the group is paused, the data is discarded at the local node.

(U//FOUO) The application joins an available multicast group using the *Join Multicast Group* command. Once it joins the multicast group, the application begins receiving data from the group, where the source is identified by the multicast peer index returned by the join command. The application may also send data to the joined multicast group, again using the peer index returned by the join command.

(U//FOUO) The application may stop a started multicast group or leave a multicast group using the *Leave Multicast Group* command. If the originator stops a multicast group, all nodes that have joined the group leave the group automatically, and the application receives a *Multicast Activity Indication* indicating the group has been left.

9.3.3.3 (U) Broadcast Stream

(U//FOUO) Each node automatically establishes a broadcast stream at power-up. The application may issue a broadcast data packet by using the peer index reserved for broadcast data.

9.3.4 (U) Firmware Update

(U//FOUO) The ISW node stores firmware in flash. The flash contains two firmware banks: active and standby. The node executes the firmware in the active slot. The application may retrieve the version of either the active or standby image using the *Get Image Version* packet.

(U//FOUO) At any time after power-up, the application may download new firmware to the ISW node. The application loads firmware by issuing a serial sequence of *Load Image* packets. The ISW node stores this image in the standby location of flash. Once the image is loaded, the application issues a *Run Image* packet to cause the new image to become the active image. However, the new image does not take effect until the application performs a reset. The reset can be performed by power cycling the ISW node or by issuing a *Reset* packet. The sequence, then, is:

- (U//FOUO) Application issues *Get Image Version* for active slot, then verifies that the new image is actually different from the old image.
- (U//FOUO) Application issues *Load Image* packet number 1. Node returns response.
- (U//FOUO) Application issues *Load Image* packet number 2 ... N – 1. Node returns response.
- (U//FOUO) Application issues *Load Image* packet number N, which contains the image label. Node returns response.
- (U//FOUO) If all responses indicate success, application issues *Run Image* for standby slot. Node returns response.
- (U//FOUO) Application issues *Reset*. Node resets.

9.3.5 (U) FCC System Certification

(U//FOUO) FCC system certification requires the measurement of unintentional emitters in a system. The ISW node's PHY is an intentional emitter, so it must be disabled during FCC testing. To disable all wireless emissions, the application must issue an *Idle Radio* packet to the node. To resume normal operation, the application can power cycle the node, issue a *Reset Radio* packet, or issue a *Reset* packet.

10. (U) SolNet Integration

(U//FOUO) The basic ISW mechanism used for identifying a class of devices is device type. However, this mechanism is not extensible and does not distinguish between different devices with same device type. For example, the device type TWS (thermal weapon sight) does not distinguish between different kinds of sights that emit differently encoded video. To bridge that gap, the ISW Protocol Specification describes the ISW Networking Protocol (SolNet).

(U//FOUO) SolNet consists of a set of over-the-air peer-to-peer commands used for discovering and subscribing to services. Some commands are mandatory, others are optional. SolNet also describes a tool chest of descriptors that can be used to describe provided services. These include vendor-defined fields to allow integrators flexibility in defining descriptors. Integrators are expected to create descriptors that suitably describe their application, and work with other integrators to ensure peer-to-peer interoperability.

(U//FOUO) SolNet nodes fall into three categories:

- (U//FOUO) Nodes that provide and consume services.
- (U//FOUO) Nodes that only consume services.
- (U//FOUO) Nodes that only provide services.

(U//FOUO) In the context of ISW, a service generally correlates to the sending of data from the offering device, and often the receiving of data associated with controlling the service.

(U//FOUO) For ISW compliance, the following table shows the commands and responses that are required for nodes that consume services versus nodes that only offer services. The columns denote if a node must be able to create and/or parse the various messages. The table shows consumer versus provider

requirements. A node that consumes and provides services must support all of the required messages for consumer and provider columns.

Table 10-1: (U) ISW-Compliant SolNet Command/Response Support

(U//FOUO)

	Consumer		Provider	
	Create	Parse	Create	Parse
Discovery Messages				
Browse Service Request	<i>Required</i>	<i>Required</i>	<i>Optional</i>	<i>Required</i>
Service Advertisement	<i>Required</i>	<i>Required</i>	<i>Required</i>	<i>Optional</i>
Capture Timestamp	<i>Optional</i>	<i>Optional</i>	<i>Optional</i>	<i>Optional</i>
Flow Messages				
Get Status Request	<i>Required</i>	<i>N/A</i>	<i>N/A</i>	<i>Required</i>
Get Status Response	<i>N/A</i>	<i>Required</i>	<i>Required</i>	<i>N/A</i>
Register Request	<i>Required</i>	<i>N/A</i>	<i>N/A</i>	<i>Required</i>
Register Response	<i>N/A</i>	<i>Required</i>	<i>Required</i>	<i>N/A</i>
Deregister Request	<i>Required</i>	<i>N/A</i>	<i>N/A</i>	<i>Required</i>
Deregister Response	<i>N/A</i>	<i>Required</i>	<i>Required</i>	<i>N/A</i>
Revoke Registration Request	<i>N/A</i>	<i>Required</i>	<i>Optional</i>	<i>N/A</i>
Revoke Registration Response	<i>Required</i>	<i>N/A</i>	<i>N/A</i>	<i>Optional</i>
Poll Data Request	<i>Required</i>	<i>N/A</i>	<i>N/A</i>	<i>Required</i>
Poll Data Response	<i>N/A</i>	<i>Required</i>	<i>Required</i>	<i>N/A</i>
Autonomous Start/Stop Request	<i>Required</i>	<i>N/A</i>	<i>N/A</i>	<i>Required</i>
Autonomous Start/Stop Response	<i>N/A</i>	<i>Required</i>	<i>Required</i>	<i>N/A</i>

(U//FOUO)

(U//FOUO) A particular device may expect to consume only particular services. Each service descriptor includes a Service ID. A service may contain child services. The SolNet descriptor topography allows the consumer to skip or ignore unknown or unwanted services. When parsing a Service Advertisement, the consumer may quickly decide whether or not the device offers services it needs. The device need only establish a flow with the services it needs.

(U//FOUO) To extend the concept of device type, the integrator may use the Label service descriptor. When applied at the top level, the 16-byte Label descriptor may specifically identify a product. For example, "FWS-I", "FWS-CS", "SBP ENVG III", or "SBP ENVG-B" distinguish among specific types of weapons sights and smart battery packs for existing programs. The same Label descriptor applied as a child descriptor can identify specific sub-services of an integrated device. For example, "DAGR GPS", "GSSIP-153", "IMU A", or "IMU B" distinguish among specific types of positioning devices or positioning protocols. This flexibility allows new programs to identify new devices and services without needing to extend the ISW Protocol Specification.

11. (U) Other Peripheral Interfaces

(U) Each module supports two additional peripheral interfaces:

- (U//FOUO) LED
- (U//FOUO) PPS (Pulse Per Second)

11.1 (U) LED

(U//FOUO) The LED resides on the module. Under normal conditions the LED is turned off. The LED lights when the application performs network association. When in association invite mode, the LED blinks once per second until association completes or times out. When in association search mode, the LED stays on steadily until association completes or times out.

11.2 (U) PPS

(U//FOUO) A single I/O line is available for pulse per second operation. A PPS signal may be fed to the pin as input, or a PPS signal may be obtained from the pin. The pin operates as follows:

- (U//FOUO) After power up, the pin remains idle.
- (U//FOUO) When the node completes its first connection to a peer, the node samples the pin for two seconds. If it detects a pulse, the pin becomes a PPS input. If it does not detect a pulse, it becomes a PPS output.

(U//FOUO) Because the node only samples on first connection, a PPS input signal must be present at that time for a node to operate as a PPS source.

(U//FOUO) When operating as a PPS output, the pin only pulses if a peer is serving as a PPS source. Any given network can only have one PPS input. The output pulse is 20 microseconds in length, and pulses with a jitter of +/- 4 microseconds.

(U//FOUO) Refer to the Physical Integration section for more information about the PPS pin.

12. (U) Antenna Integration

(U//FOUO) For any system, the antenna selection and antenna placement have a greater effect on performance than any other factor. For this reason, the integrator should involve the module vendor during mechanical design to determine antenna selection and antenna placement.

12.1 (U) Antenna Selection

(U//FOUO) All modules include a u.fl connector. The USB-attached module also supports an on-board chip antenna. A 0 ohm resistor is used to enable the chip antenna or the u.fl connector. The integrator must determine if the on-board antenna is sufficient and appropriately placed. If not, the integrator must use the u.fl connector to attach an external antenna.

(U) When considering external antennas, UWB supports three antenna types:

- (U//FOUO) Dipole (whip)
- (U//FOUO) Chip
- (U//FOUO) Patch

(U//FOUO) Because UWB covers such a wide range of frequencies, omnidirectional antenna design across that range is difficult. There are a number of antennas on the market that claim UWB coverage. However, they differ widely in consistency across the frequency range. In particular, a number of

antennas perform well in the lower bands (Bandgroup 1) but become directional in the upper bands (Bandgroup 6). Consult the module vendor to determine which antennas are available of each type and how well each performs across the entire frequency range.

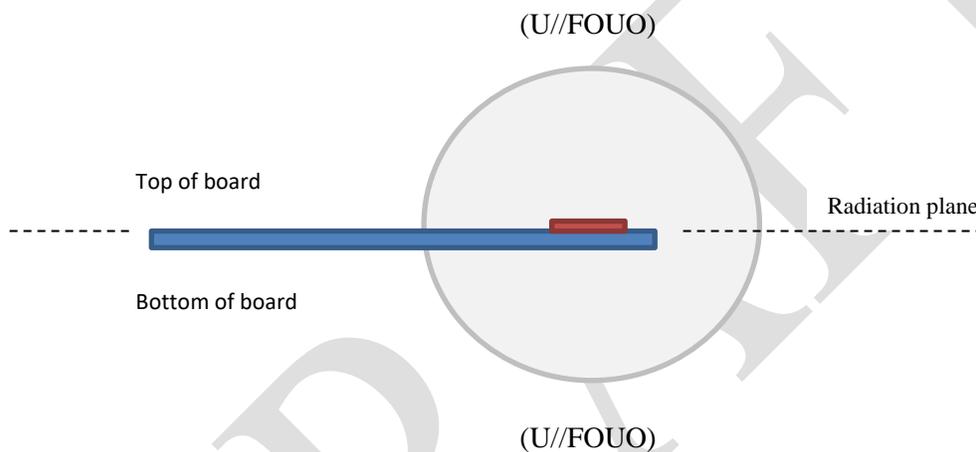
12.2 (U) Antenna Placement

(U//FOUO) Antenna placement concerns three properties: location in a device, loss induced by connections, and orientation of the antenna.

12.2.1 (U) Antenna Location

(U//FOUO) The integrator must place the antenna where it is the least obstructed by surrounding metal. For example, consider the placement of a chip antenna in a system. The diagram below represents the side view of a board with the chip antenna at one end, and the expected radiation pattern (circle) and radiation plane. The radiation plane represents the bottom of the antenna chip.

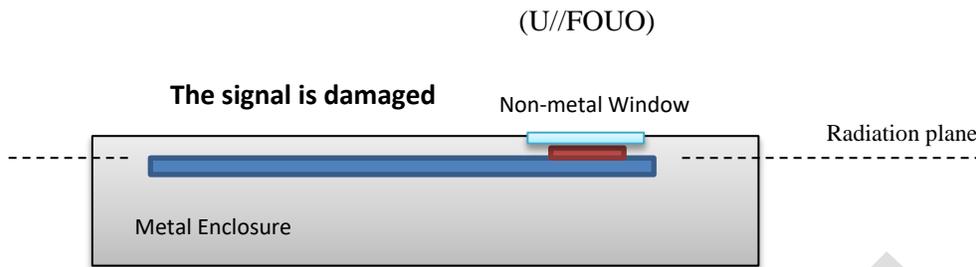
Figure 12-1: (U) Antenna Radiation Plane



(U//FOUO) When the board resides in plastics or other non-metallic enclosures, the UWB signal radiates uniformly from the enclosure. When fully enclosed in metal, the UWB signal is blocked by the metal. In order for the antenna to function, the bottom of the chip antenna (shown as "radiation plane") must sit at least 1 mm above the top plane of the metal enclosure. Otherwise, reflections from the metal damage the signal.

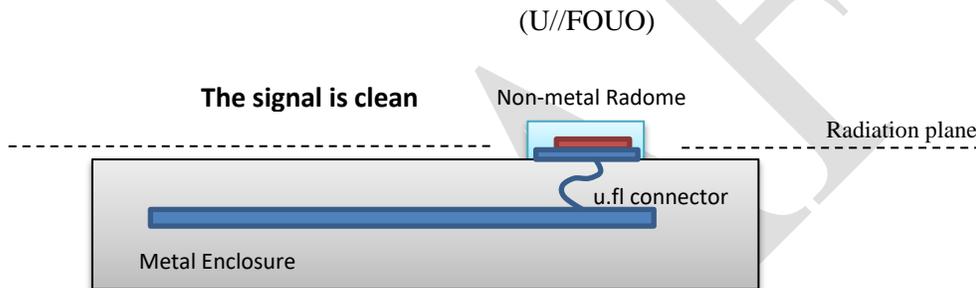
(U//FOUO) For example, if the board is placed at the top of the metal enclosure with a plastic window for the antenna, the UWB signal is still damaged because the radiation plane is still below the highest plane of the metal container.

Figure 12-2: (U) Improper Antenna Placement



(U//FOUO) The diagram below shows a viable alternative. A separate antenna board attached to the radio board via a u.fl connector sits with its radiation plane above the highest plane of the metal enclosure. The bottom side of the antenna board may sit on the metal enclosure as long as the bottom of the chip antenna (radiation plane) is 1 mm above the highest plane of the metal enclosure.

Figure 12-3: (U) Proper Antenna Placement



(U//FOUO) For a three-dimensional device, the integrator must determine where on the device to place the antenna. To do so, the integrator must decide which side is generally the least impeded and/or has the best line of sight to other radios in the system. For example, if a device is worn on the body, the antenna should reside on a surface facing away from the body.

(U//FOUO) Another strategy for antenna placement is to separate the antenna from the device using a cable. This allows the module to be embedded inside a device surrounded by metal, and the antenna is brought outside of the metal. The integrator may elect to bring the antenna away from the device using a longer cable. For example, a device may be worn on the body in an obstructed way, but the antenna may be pulled out to a better location, such as the shoulder. Refer to antenna connection loss for caveats.

12.2.2 (U) Antenna Connection Loss

(U//FOUO) All antenna connections result in UWB signal loss. The loss depends on the following factors:

- (U//FOUO) The length of a connecting cable. The longer the cable, the more the loss. The integrator should minimize cable length.
- (U//FOUO) The loss characteristics of the connecting cable. The loss characteristics of cables vary significantly. The integrator should use cable suitable for UWB signals with the lowest possible loss.

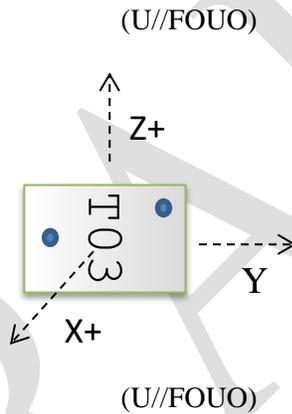
- (U//FOUO) The number of connection points along the antenna path. Each connection point introduces loss. The vendor should minimize the number of connection points.
- (U//FOUO) The loss characteristics of each connection along the antenna path. Each connection point must be carefully impedance matched to avoid additional loss. The integrator must ensure precise impedance matching to 50 ohms.

(U//FOUO) The UWB module vendor can boost the signal somewhat to account for antenna connection loss. This typically happens during module calibration. In particular, the vendor may calibrate the module through the particular antenna cable and connection sequence used by the actual device. Contact the UWB module vendor for details.

12.2.3 (U) Antenna Orientation

(U//FOUO) No antenna is perfectly omnidirectional. As a result, the orientation of the antenna in the system can affect system performance. For example, consider the TRDA chip antenna in the following figure.

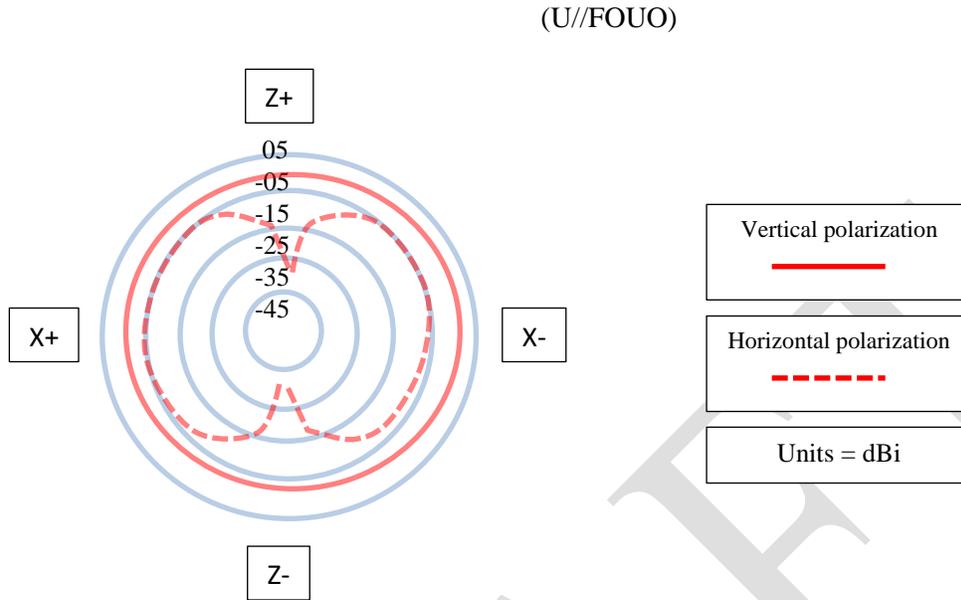
Figure 12-4: (U) TRDA Chip Antenna



(U//FOUO) The chip radiates in three planes: X, Y, and Z.

(U) The following figure shows the X/Z radiation pattern at a frequency in Bandgroup 1.

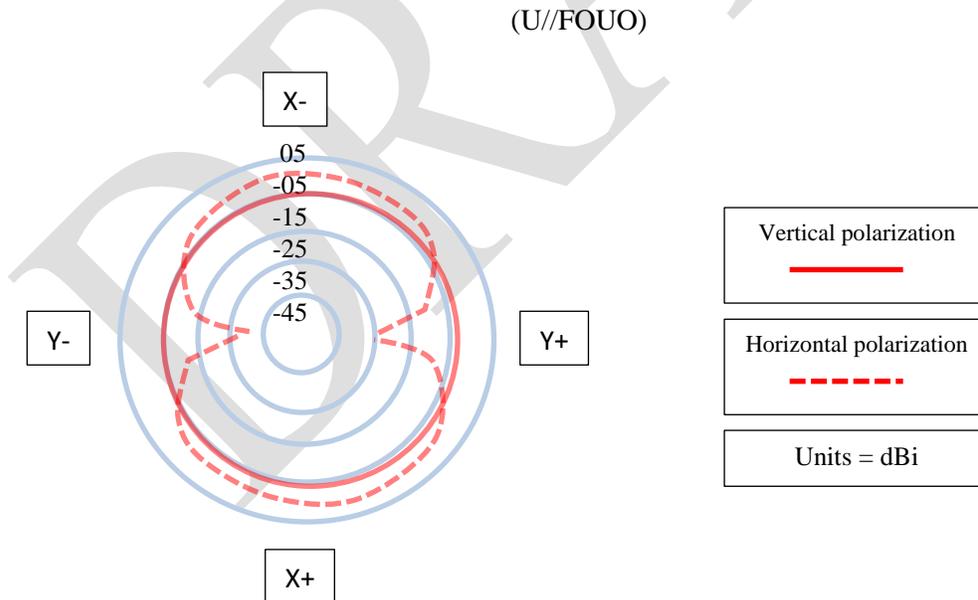
Figure 12-5: (U) X/Z Radiation Pattern at 3.96 GHz.



(U//FOUO)

(U//FOUO) The following figure shows the X/Y radiation pattern at the same frequency in Bandgroup 1.

Figure 12-6: (U) X/Y Radiation Pattern at 3.96 GHz



(U//FOUO)

(U//FOUO) These diagrams show that the X plane is more uniform than the Y and Z planes. This indicates that the flat face of the antenna (X) should face other antennas in the system as best as possible.

Appendix A. (U) SolNet Descriptor Examples

A.1. (U//FOUO) Weapon Sight to Heads-up-Display

(U//FOUO) In this example, a weapon sight (WS) sends video and inertial motion data to a heads-up-display (HUD). The WS and HUD also exchange control packets belonging to an integrator-defined protocol. Assume that the makers of the WS and HUD agree on the following:

- (U//FOUO) Video uses Endpoint 1
- (U//FOUO) Inertial motion data uses endpoint 2
- (U//FOUO) Control data uses Endpoint 3

(U//FOUO) Note: Endpoint 0 is used for SolNet protocol that performs the discovery and registration.

(U//FOUO) The video is 8-bit/pixel monochrome with a resolution of 640x480 at 30 frames per second. The inertial motion data is in a fictional standard format we will call "IMU-2543". The control data consists of protocol packets belonging to fictional standard we will call "Pretty Good Device Protocol Rev 1" (also called "PGDP-R1").

(U//FOUO) The WS must use a service descriptor to describe the data it provides, which means video, inertial motion data, and control data. The HUD must use a service descriptor to describe the data it provides, which means control data. Both devices also use service descriptors to describe the name of devices and services. For reference, let's call our particular weapon sight "WS-NextGen-2" and our particular heads-up-display "HUD-Bino-1".

A.1.1 (U//FOUO) Weapon Sight Descriptor Set

(U//FOUO) To encode the WS services, define four service descriptors:

- (U//FOUO) Label service: Describes the overall device
- (U//FOUO) Video service: Describes the video output
- (U//FOUO) Position service: Describes the inertial motion output
- (U//FOUO) Protocol service: Describes the control data input/output

(U) The Label service descriptor is shown in the following table.

Table 12-1: (U) WS Label Descriptor

(U//FOUO)

Field	Bits	Value	Description
Length	16	0x0C	Length of this descriptor
Service ID	16	0x0100	Label Descriptor
Label Text	96	"WS-NextGen-2"	Label

(U//FOUO)

(U//FOUO) Note: Vendors may determine that the top-level label descriptor is sufficient to understand the capabilities of a particular peer. In other words, the browsing device might know explicitly all the services of a device advertising itself as "WS-NextGen-2". However, to allow for flexibility in the field and changes to devices, the vendor may include the following descriptors to explicitly describe the services.

(U//FOUO) The Video service descriptor contains two child descriptors: an endpoint descriptor and a label descriptor. The combined descriptor is shown in the following table.

Table 12-2: (U) WS Video Descriptor Set

(U//FOUO)

Field	Bits	Value	Description
Length	16	0x0022	Length of this descriptor and all children
Service ID	16	0x0001	Video service
Service selector	32	0x00000001	Opaque handle known by the sender
Stream format	16	0x0000	Uncompressed video
Bandwidth	8	0xCA	74 Mbps
Latency	8	0x03	12 milliseconds
Length	16	0x0004	Length of this child descriptor
Service ID	16	0x0102	Endpoint Descriptor
Endpoint ID	8	0x01	Provide data on endpoint 1
Endpoint Distribution	8	0x00	Unicast
Data Priority	8	0x01	High priority
Reserved	8	0x00	<i>Reserved for future definition</i>
Length	16	0x0E	Length of this child descriptor
Service ID	16	0x0100	Label Descriptor
Label Text	112	"640x480x30Mono"	Label

(U//FOUO)

(U//FOUO) The Position service descriptor contains two child descriptors: an endpoint descriptor and a label descriptor.

(U) The combined descriptor is shown in the following table.

Table 12-3: (U) WS Positioning Descriptor Set

(U//FOUO)

Field	Bits	Value	Description
Length	16	0x0020	Length of this descriptor and all children
Service ID	16	0x0003	Positioning service
Service selector	32	0x00000002	Opaque handle known by the sender
Datum Array	64	0x0607080000000000	Measures acceleration, angular velocity, flux density
Length	16	0x0004	Length of this child descriptor
Service ID	16	0x0102	Endpoint Descriptor
Endpoint ID	8	0x02	Provide data on endpoint 2
Endpoint Distribution	8	0x00	Unicast
Data Priority	8	0x00	Low priority (best effort)
Reserved	8	0x00	<i>Reserved for future definition</i>
Length	16	0x08	Length of this child descriptor
Service ID	16	0x0100	Label Descriptor
Label Text	64	"IMU-2543"	Label

(U//FOUO)

(U//FOUO) The Protocol service descriptor contains two child descriptors: an endpoint descriptor and a label descriptor. The combined descriptor is shown in the following table.

Table 12-4: (U) WS Protocol Descriptor Set

(U//FOUO)

Field	Bits	Value	Description
Length	16	0x0017	Length of this descriptor and all children
Service ID	16	0x0007	Protocol service
Protocol Identifier	16	0x0000	Unspecified
Reserved	16	0x0000	Reserved
Length	16	0x0004	Length of this child descriptor
Service ID	16	0x0102	Endpoint Descriptor
Endpoint ID	8	0x03	Provide data on endpoint 3
Endpoint Distribution	8	0x00	Unicast
Data Priority	8	0x00	Low priority (best effort)
Reserved	8	0x00	<i>Reserved for future definition</i>
Length	16	0x08	Length of this child descriptor
Service ID	16	0x0100	Label Descriptor
Label Text	56	"PGDP-R1"	Label

(U//FOUO)

(U//FOUO) The Service Advertisement packet for putting these descriptors on air is shown in the following table.

Table 12-5: (U) WS Service Advertisement

(U//FOUO)

Field	Bits	Value	Description
Protocol Version	8	0x00	Protocol Version
Protocol Class	8	0x01	SolNet protocol
Message Class	8	0x01	Discovery protocol
Message ID	8	0x81	Service Advertisement
Sequence Number	8	Var	Incrementing count
Status	8	0x00	Non-error status
Checksum	16	Var	Checksum value
Message Length	32	0x75	Length of payload following this field
Label Service Descriptor	128	Var	Label service descriptor set
Video Service Descriptor	304	Var	Video service descriptor set
Positioning Service Descriptor	288	Var	Positioning service descriptor set
Protocol Service Descriptor	216	Var	Protocol service descriptor set

(U//FOUO)

A.1.2 (U//FOUO) Heads-Up-Display Descriptor Set

(U//FOUO) To encode the HUD services, define two service descriptors:

- (U//FOUO) Label service: Describes the overall device
- (U//FOUO) Protocol service: Describes the control data input/output

(U//FOUO) The Label service descriptor is shown in the following table.

Table 12-6: (U) HUD Label Descriptor

(U//FOUO)

Field	Bits	Value	Description
Length	16	0x0C	Length of this descriptor
Service ID	16	0x0100	Label Descriptor
Label Text	96	"HUD-Bino-1"	Label

(U//FOUO)

(U//FOUO) The Vendor service descriptor contains two child descriptors: an endpoint descriptor and a label descriptor. The combined descriptor is shown in the following table.

Table 12-7: (U) HUD Protocol Descriptor Set

(U//FOUO)

Field	Bits	Value	Description
Length	16	0x0017	Length of this descriptor and all children
Service ID	16	0x0007	Protocol service
Protocol Identifier	16	0x0000	Unspecified
Reserved	16	0x0000	Reserved
Length	16	0x0004	Length of this child descriptor
Service ID	16	0x0102	Endpoint Descriptor
Endpoint ID	8	0x03	Provide data on endpoint 3
Endpoint Distribution	8	0x00	Unicast
Data Priority	8	0x00	Low priority (best effort)
Reserved	8	0x00	<i>Reserved for future definition</i>
Length	16	0x08	Length of this child descriptor
Service ID	16	0x0100	Label Descriptor
Label Text	56	"PGDP-R1"	Label

(U//FOUO)

(U//FOUO) The Service Advertisement packet for putting these descriptors on air is shown in the following diagram.

Table 12-8: (U) HUD Service Advertisement

(U//FOUO)

Field	Bits	Value	Description
Protocol Version	8	0x00	Protocol Version
Protocol Class	8	0x01	SolNet protocol
Message Class	8	0x01	Discovery protocol
Message ID	8	0x81	Service Advertisement
Sequence Number	8	Var	Incrementing count
Status	8	0x00	Non-error status
Checksum	16	Var	Checksum value
Message Length	32	0x2B	Length of payload following this field
Label Service Descriptor	128	Var	Label service descriptor set
Protocol Service Descriptor	216	Var	Protocol service descriptor set

(U//FOUO)

Appendix B. (U//FOUO) FWS to ISW Porting Guidance

B.1. (U) Overview

(U//FOUO) FWS systems implemented a pre-ISW variation of protocol. This appendix describes the impact to the Mac-ISW client (host system, application) for porting from the legacy approach to the ISW approach.

B.2. (U) Wire Interface

(U//FOUO) The wire interfaces are fundamentally the same as those of the legacy system. However, there are a few ISW-related modifications that impact software and potentially hardware for the parallel-attached system.

B.2.1 (U) Word-Aligned API Enforcement

(U//FOUO) All API commands must be 32-bit word-aligned when sent over the wired interface by the application. Non-word aligned commands are discarded. Some legacy systems would issue certain commands non-word aligned. NIST fuzz testing requires a strict enforcement of the API spec versus what is sent, so word alignment is now strictly enforced.

B.2.2 (U) Peer Indexing Change

(U//FOUO) Peer indexing has changed from legacy systems. For parallel-attached systems, peer indexes 0x00 – 0x0E were used for sending data to peers on endpoints 1 – 3, while peer index 0x00 / endpoint 0 was used to send API commands to the local node. For USB and serial-attached system, peer indexes

0x00 – 0x0E were used for sending data to peers on endpoints 0 – 3, while peer index 0x1F / endpoint 0 was used to send API commands to the local node.

(U//FOUO) ISW expands the peer indexing. The following table, copied from the ISW Protocol Specification, shows the peer new indexing.

Table 12-9: (U) Peer Indexes

(U//FOUO)

Peer Index Range	Description
0x00 – 0x1E	Application-level data targeting peers 0 through 31. These peer indices map directly to those returned by the <i>Streams Status Indication</i> and <i>Get Streams Status</i> .
0x1F	Application-level data targeting the local application. This is essentially a data loopback.
0x20 – 0x3E	Control data targeting a peer node. Used for sending ISW API commands to peers.
0x3F	Control data targeting the local node. Used for sending ISW API commands to the local node. Data targeting this peer index should always use endpoint number 0.
0x40 – 0x5E	Application-level data targeting a multicast group.
0xFF	Application-level data broadcast to all peers.

(U//FOUO)

(U//FOUO) For ISW, all module types now use the same indexing. The primary impact for legacy systems is that API commands now target peer index 0x3F instead of 0x1F or 0x00 as before, depending on module type. For USB and serial-attached modules, the peer index and endpoint are specified in a SWIM header that precedes the packets. For parallel-attached modules, the peer index and endpoint are encoded in the GP_HOST and GP_MAC registers as they were before (no SWIM header). However, the definition of the GP_HOST and GP_MAC register is modified to allow for the expanded peer index value.

(U) The following table, copied from the Alereon Simplified Parallel Bus Operation, shows the new format.

Table 12-10: (U) GP_HOST and GP_MAC Register, New
(U//FOUO)

Bits (lsb = 0)	Description
0 - 14	Data Length: Length in bytes of the packet to be read or written. Maximum length = 4064.
15 - 16	Reserved
17	Backpressure Marker: <ul style="list-style-type: none"> GP_MAC – Version 1: If set, all non-zero Endpoints are backpressured against writes; any packets to non-zero Endpoints are discarded by the Device. Version 2: If set, all data is backpressured against writes except for data from Peer Index 0x1F; packets to all other Peer Indices are discarded by the Device. GP_Host – Not used.
18-19	Reserved
20-23	Peer Index msb: Most significant bits of index of the peer to which packet belongs.
24-27	Endpoint: Number of the virtual endpoint to which the packet belongs.
28-31	Peer Index lsb: Least significant bits of index of the peer to which packet belongs.

(U//FOUO)

(U//FOUO) Bits 20 – 23, which were formerly reserved, now specify the upper 4 bits of the 8 bit peer index. Bits 28 – 31 still contain the lower 4 bits of the peer index.

B.3. (U) Initialization and Reset

B.3.1 (U) Autonomous Indications

(U//FOUO) On some legacy systems, indications (reset, connection, association, streams status) were automatically suppressed at power-up. The application was forced to enable indications.

(U//FOUO) For ISW, indications are automatically enabled from power-up. The application no longer needs to enable them.

B.3.2 (U) Reset Indication and Start-up Sequence

(U//FOUO) On legacy systems, there was no *Reset Indication*. After bus enumeration, the application could begin sending messages as soon as it was ready. For parallel and serial-attached systems, the lack of *Reset Indication* could mask an under-the-covers reset, leaving certain application-set parameters unset.

(U//FOUO) For ISW, a *Reset Indication* is issued after every power-up and after every hard reset. If the reset fails due to internal hardware failure, the *Reset Indication* will include a result code of non-zero. This indicates that the module will reset again in one second. However, under normal operation, the *Reset Indication* result code is 0, meaning success. As soon as the application receives a successful reset indication, it must perform a security login and start cryptographic operations. Failure to do this immediately will delay the connection times of the module. The command to log in is *Authenticate Role*, and the command to start cryptography is *Crypto Session Start*. These operations are required for NIST certification.

(U//FOUO) Once the application logs in and starts the cryptographic operations, the application must check and set parameters as it did in the legacy systems. If the application receives a *Reset Indication* not related to power up, it must check and set parameters again.

B.4. (U) Device Identity

(U//FOUO) On legacy systems, each device type had a hard-coded behavior in terms of whether or not it could be a coordinator (master). The device type / role characteristic varied among vendors.

(U//FOUO) For ISW, the role has been divorced from the device type. The ISW Protocol Specification section 6.3.1.1 defines the default role for each device type. However, the application can (and should) override the default using a Set Coordinator command and persist it in NVRAM on the module. This allows the vendor to use any device type to reasonably represent the device, and then set the role as coordinator-capable or not coordinator-capable and set the coordinator priority.

B.5. (U) Network Association

(U//FOUO) Legacy systems used pairwise association, forcing a device to associate with all other devices with which it needed to converse. Although not an issue for FWS, this resulted in a large number of point-to-point association operations for networks with more nodes. Regardless, every association was initiated by the issuing of a generic *Start Association* command on both nodes that needed to associate with one another.

(U//FOUO) ISW implements a new association model where nodes may join a network by associating with a single node already on that network. After associating with one node, the joining node can converse with all nodes on the network. To support this new model, ISW adds a type field the *Start Association* command. When adding a device to a network, the application on the node that is already part of the network issues a *Start Association* command with type = invite. The application on the node that is joining the network issues a *Start Association* command with type = search. Upon successful completion, both nodes return the same *Association Indication* as before.

(U) Caveats:

- (U//FOUO) If starting with two unassociated nodes and both are coordinator-capable, it does not matter which one is the joiner and which is the searcher.
- (U//FOUO) If starting with two unassociated devices where one is coordinator-capable and the other is not, the coordinator-capable device must be the inviter.
- (U//FOUO) After the non-coordinator capable device joins a network, it may invite other devices to the network as long as the network coordinator is still on air.

B.6. (U) Idle Scan Frequency

(U//FOUO) On legacy systems, the idle scan frequency was 8 bits. This resulted in a max idle scan frequency period of 255 seconds. For certain kinds of testing, the idle scan frequency needed to be effectively disabled, and there was no mechanism to do so.

(U//FOUO) ISW expands the idle scan frequency to 16 bits. This allows the frequency to effectively be disabled.

B.7. (U) New Capabilities

(U//FOUO) ISW implements the following new optional capabilities:

- (U//FOUO) Extended Streams Status. This adds additional information to the Streams Status messages, including MAC address, channel, etc. This information may be useful to the

application when handling a large number of nodes. Streams Status now includes multiple versions; version 2 is the recommended default. The new status is shown in the *Streams Status Indication*. A new command, *Get Extended Streams Status*, gets a snapshot of the new status. The *Set Indications* command allows the application to specify which version to return autonomously.

- (U//FOUO) Quality of Service. QoS may be optionally defined per stream and substream. Per stream, the application may cap the latency. Per substream, the application may define the substream priority and whether or not the substream contents are discardable under congestion conditions.
- (U//FOUO) Multicast / Broadcast. The application may now broadcast and multicast data. To support multicast, a number of new commands have been added to start, join, leave, and monitor multicast groups. The application may start one multicast group and/or join multicast group simultaneously.
- (U//FOUO) Power Management. The application may cause the radio to turn off without cutting power to the module using a *Radio Idle* command. This saves power. The application may resume the radio by issuing a *Radio Reset* command.

B.8. (U) Discovery and Registration (SolNet)

(U//FOUO) Legacy systems used the device type to make assumptions about what device was connected and what kind of data it provided/consumed on which endpoint. However, the device type alone is not sufficient to distinguish among variations of devices. For example, the device type TWS (thermal weapon sight) is not sufficient to distinguish between FWS-I and FWS-CS.

(U//FOUO) ISW adds the SolNet protocol as a standard means of determining the details of a device, registering for its data (services), and starting the flow of data. ISW nodes must implement the specified subset of SolNet to allow for future expansion of networks. Appendix A provides an example of a set of descriptors for a weapon sight and heads-up-display.