Overview

The RF Communication Link Test Set is used to perform lab and field testing of SG90 compatible sonobuoys. A block diagram of the Test Set is shown in Figure 1. The Test Set includes three (3) VHF SG90 uplink receivers and one (1) UHF SG90 downlink transmitter. The SG90 Uplink Receiver and Downlink Transmitter are based on the KTS Wireless Agility family of radios (AxRs). The radios implement the Physical Layer functions of the SG90 standard while the SOC implements the rest. This document describes the network interface between the radios and the System Operator Console (SOC).

This network interface must support two primary functions: 1) Radio Management (configuration and monitoring) of the individual receivers and transmitters, and 2) Traffic Management (sending and receiving SG90 Data Link Layer PDUs).

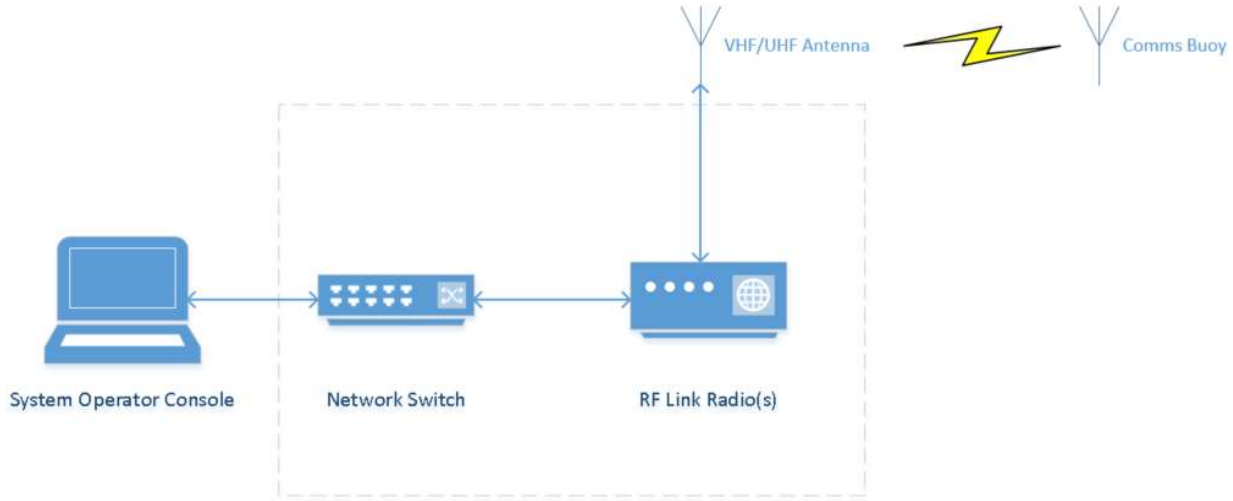

Figure 1 - RF Communication Link Test Set

Radio Management

The AxRs are managed through a Management Data Base (MDB). The MDB is composed of MDB Variables which define things like the radio's center frequency or transmit power level. MDB Variables are defined to be 16 bits in length, but multiple variables can be combined to create wider values that are multiples of 16 bits. For instance, IPV4 addresses are 32 bit values so two 16 bit MDB variables are combined to create the 32 bit address. In all cases where multiple MDB variables are combined to create a managed variable, the variable is updated when the MDB variable containing the most significant bits is written.

The MDB is accessed using Modbus-TCP over the Ethernet interface. The KTS Wireless Element Management Software tool (EMS) which is provided with the radios uses this mechanism.

Among other things, the Modbus protocol defines a mechanism to access 16-bit registers. In the AxR, there is a one-to-one correspondence between these registers and the 16-bit words that make up the MDB. So, monitoring status of the radio involves performing a Modbus read of the corresponding MDB Variable. And, changing the configuration of the radio involves performing a Modbus write of the corresponding MDB variable. For this reason, the AxR supports two Modbus Function Codes:

- 0x03 – Read Holding Registers
- 0x06 – Write Single Register

The payload of these Modbus messages includes a register address. The register address is set to (MDB Index – 1), where the MDB Index is defined in Table 1.

What is the list of things Sparton/Raytheon wants to modify or monitor?

MDB Index	Variable Name	Variable Description
2	bitrateLo	Radio Data Rate in units of bits per second. The value is a 32 bit unsigned integer with the upper portion in bitrateHi and the lower portion in bitrateLo.
3	bitrateHi	See bitrateLo
4	centerFreqLo	The Radio Center Frequency in units of Hz. The value is a 32 bit unsigned integer with the upper portion in centerFreqHi.
5	centerFreqHi	See centerFreqLo
14	fecMode	TBD
15	txPower	Radio Transmit Power Level in tenths of a dB. The value is a 16 bit unsigned integer in the range of [100, 370]
22	rssi	Received Signal Strength of the last received burst or block in dB. The value is a 16 bit signed integer.
24	temp	The internal temperature of the radio in units of degrees C. The value is a 16 bit signed integer.
25	Alarm	This is a 16-bit bitfield where each bit represents a different alarm condition
27	ipAddressLo	ipAddressLo and ipAddressHi are combined to create the static IP address assigned to the RJ45 port of the radio. The IP address is stored in network order
28	ipAddressHi	
29	netMaskLo	netMaskLo and netMaskHi are combined to create the static subnet mask
30	netMaskHi	

31	gatewayLo	gatewayLo and gatewayHi are combined to create the static gateway address
32	gatewayHi	
38	broadcastAddrLo	broadcastAddrLo and broadcastAddrHi are combined to create the static broadcast address
39	broadcastAddrHi	
40	serverAddrLo	serverAddrLo and serverAddrHi are combined to create the static IP address of the SOC server.
41	serverAddrHi	
44	networkRestart	Any write to this variable causes the radio to start using the latest changes to the static IP network addresses and masks.
52	presetConfigs	This 16 bit unsigned integer defines a collection of preset configurations. Valid configurations are TBD.
53	resetBoard	Any write to this variable will force the radio to perform a soft reset.

Table 1 - AxR MDB Variables

Traffic Management

The Test Set must generate SG90 downlink traffic, and receive and process SG90 uplink traffic. The traffic interface between the SOC and the AxRs is implemented using a UDP socket on port 3123.

SG90 Downlink Transmit Interface

The AxR will perform all the Physical layer functions as defined by section 5.1 of the SG90 specification. The SOC is responsible for the Downlink MAC Protocol functions defined by section 5.2. So, the interface between the SOC and the radio is the MPDU defined by section 5.2.1. In particular, all the bytes between Aircraft Address and CRC16, inclusive, shown in Figure 20 of the SG90 spec are sent to the AxR in a single UDP packet.

The SOC must prepend a single byte containing the value 0x01 to the MPDU as part of the UDP packet. This byte indicates this MPDU is to be transmitted. The radio removes this byte prior to transmission.

The radio will not perform any error checking on the UDP packet payload. It will simply take the data from the UDP payload, format it according to the SG90 spec and transmit it in a single downlink burst.

SG90 Downlink Receive Interface

The AxR will receive the SG90 downlink transmission and extract the MPDU as defined by section 5.2.1. It will prepend a single byte with the value 0x02 indicating this is a received MPDU. The MPDU with the prepended byte will be sent to the SOC in a single UDP packet. The AxR will use the IPV4 address configured in MDB variables 40/41 as the destination address.

The radio will not perform any error checking on the MPDU.

SG90 Uplink Transmit Interface

The AxR will automatically start transmitting Preamble Frames as defined in the SG90 spec. Non-preamble frames are sent to the radio over a UDP packet. The SOC sends a complete DLL frame as defined in section 4.1.2 (all the bytes between the Format ID and the last byte of the Application Data, inclusive) in a single UDP packet. There are two modifications required to this standard DLL frame:

1. The Format ID should be in the form of the 5-bit integer as defined in Table 3 of the SG90 spec. The radio will translate this to the 16-bit codeword as defined in Table 7.
2. A single byte with a value of 0x01 must be prepended to the DLL frame to indicate this is to be transmitted. The radio will remove this byte prior to transmission.

No error checking will be performed on the DLL frame.

SG90 Uplink Receive Interface

The AxR will receive the SG90 uplink transmission and extract DLL frames as defined by section 4.1.2. All the bytes between the Format ID and the last byte of the Applications Data, inclusive, as shown in Figure 5 of the SG90 spec are sent to the SOC in a single UDP packet. There are two modification to the data:

1. The Format ID will be translated from the 16-bit codeword (as defined in Table 7) to the 5-bit integer as defined in Table 3 of the SG90 spec.
2. A single byte with a value of 0x02 will be prepended to the DLL frame prior to UDP transmission.

Only frames with a Format ID of 1 through 5, inclusive, will be sent to the SOC over the UDP socket. Frames received with a Format ID of 0 (i.e. – Preamble Frames) will be dropped. Frames received with an invalid Format ID will be dropped.