Appendix B – RCS11 Remote Communications

B.1 Host Computer Remote Communications

Control and status messages are conveyed between the RCS11 and the host computer using packetized message blocks in accordance with a proprietary communications specification. This communication is handled by the Radyne Link Level Protocol (RLLP), which serves as a protocol ‘wrapper’ for the RM&C data.

Complete information on monitor and control software is contained in the Radyne RLLP Protocol Reference Guide.

B.1.1 Protocol Structure

The Communications Specification (COMMSPEC) defines the interaction of computer resident Monitor and Control software used in satellite earth station equipment such as modems, redundancy switches, multiplexers, and other ancillary support gear. Communication is bi-directional, and is normally established on one or more full-duplex 9600-baud multi-drop control buses that conform to EIA Standard RS-485.

Each piece of earth station equipment on a control bus has a unique physical address, which is assigned during station setup/configuration or prior to shipment. Valid decimal addresses on one control bus range from 032 through 255 for a total of up to 224 devices per bus. Address 255 of each control bus is usually reserved for the M&C computer.

B.1.2 Protocol Wrapper

The Radyne COMMSPEC is byte-oriented, with the Least Significant Bit (LSB) issued first. Each data byte is conveyed as mark/space information with two marks comprising the stop data. When the last byte of data is transmitted, a hold comprises one steady mark (the last stop bit). To begin or resume data transfer, a space (00h) substitutes this mark. This handling scheme is controlled by the hardware and is transparent to the user. A pictorial representation of the data and its surrounding overhead may be shown as follows:

S1	S2	B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	S1	S2, etc.

The stop bits, S1 and S2, are each a mark. Data flow remains in a hold mode until S2 is replaced by a space. If S2 is followed by a space, it is considered a start bit for the data byte and not part of the actual data (B_0 - B_7).

The COMMSPEC developed for use with the Radyne Link Level Protocol (RLLP) organizes the actual monitor and control data within a shell, or ‘protocol wrapper’, that surrounds the data. The format and structure of the COMMSPEC message exchanges are described herein. Decimal numbers have no suffix; hexadecimal numbers end with a lower case h suffix and binary values have a lower case b suffix. Thus, 22 = 16h = 000010110b. The principal elements of a data frame, in order of occurrence, are summarized as follows:

<SYN> - the message format header character, or ASCII sync character, that defines the beginning of a message. The <SYN> character value is always 16h.

<DATA COUNT> - the 2 Byte Data Count is the number of bytes in the <DATA> field, ranging from 0 through 509.

<SOURCE ID> - the Source Identifier defines the multi-drop address origin. Note that all nodes on a given control bus have an unique address that must be defined.

<DESTINATION ID> - the Destination Identifier serves as a pointer to the multi-drop destination device that indicates where the message is to be sent.

<FRAME SEQUENCE NUMBER> -the FSN is a tag with a value from O through 255 that is sent with each message. It assures sequential information framing and correct equipment acknowledgment and data transfers.

<OPCODE> - the 2 Byte Operation Code field contains a number that identifies the message type associated with the data that follows it. Equipment under MCS control recognizes this byte via firmware identification and subsequently steers the DATA accordingly to perform a specific function or series of functions. Acknowledgment and error codes are returned in this field. 1 Byte for the DMD5000 protocol, and 2 Bytes for the DMD15 protocol.

<DATA > - the Data field contains the binary, bi-directional data bytes associated with the <OPCODE> - The number of data bytes in this field is indicated by the <BYTE COUNT> value.

<CHECKSUM> - the checksum is the modulo 256 sum of all preceding message bytes, excluding the <SYN> character. The checksum determines the presence or absence of errors within the message. In a message block with the following parameters, the checksum is computed as shown in Table 1.

BYTE FIELD	DATA CONTENT		RUNNING CHECKSUM
<BYTE COUNT>	02h	= 00000000b	00000000b
<BYTE COUNT>	02h	= 00000010b	00000010b
<SOURCEID>	F0h	= 11110000b	11110010b
<DESTINATION ID>	2Ah	= 00101010b	00011100b
<FSN>	09h	= 00001001b	00100101b
<OPCODE>	03h	= 00000011b	00101000b
<DATA> (Byte 1)	DFh	= 11011111b	00000111b
<DATA> (Byte 2)	FEh	= 11111110b	00000101b

Table 1. Checksum Calculation Example

Thus, the checksum is 00000101b; which is 05h or 5 decimal. Alternative methods of calculating the checksum for the same message frame are:

02h + FOh + 2Ah + 09h + 03h + DFh + FEh = 305h.

Since the only concern is the modulo 256 (modulo 1 00h) equivalent (values that can be represented by a single 8-bit byte), the checksum is 05h.

For a decimal checksum calculation, the equivalent values for each information field are:
0 + 2 + 240 + 42 + 9 + 3 + 223 + 254 = 773;
773/256 = 3 with a remainder of 5.
This remainder is the checksum for the frame.
5 (decimal) = 05h = 0101b = <CHECKSUM>

B.1.3 Frame Description and Bus Handshaking

In a Monitor and Control environment, every message frame on a control bus port executes as a packet in a loop beginning with a wait-for-SYN-character mode. The remaining message format header information is then loaded, either by the M&C computer or by a subordinate piece of equipment (such as the DMD15) requesting access to the bus. Data is processed in accordance with the OPCODE, and the checksum for the frame is calculated. If the anticipated checksum does not match then a checksum error response is returned to the message frame originator. The entire message frame is discarded and the wait-for-SYN mode goes back into effect. If the OPCODE resides within a command message, it defines the class of action that denotes an instruction which is specific to the device type, and is a prefix to the DATA field if data is required. If the OPCODE resides within a query message packet, then it defines the query code, and can serve as a prefix to query code DATA.

The Frame Sequence Number (FSN) is included in every message packet, and increments sequentially. When the M & C computer or bus-linked equipment initiates a message, it assigns the FSN as a tag for error control and handshaking. A different FSN is produced for each new message from the FSN originator to a specific device on the control bus. If a command packet is sent and not received at its intended destination, then an appropriate response message is not received by the packet originator. The original command packet is then re-transmitted with the same FSN. If the repeated message is received correctly at this point, it is considered a new message and is executed and acknowledged as such.

If the command packet is received at its intended destination but the response message (acknowledgment) is lost, then the message originator (usually the M&C computer) re-transmits the original command packet with the same FSN. The destination device detects the same FSN and recognizes that the message is a duplicate, so the associated commands within the packet are not executed a second time. However, the response packet is again sent back to the source as an acknowledgment in order to preclude undesired multiple executions of the same command.

To reiterate, valid equipment responses to a message require the FSN tag in the command packet. This serves as part of the handshake/acknowledge routine. If a valid response message is absent, then the command is re-transmitted with the same FSN. For a repeat of the same command involving iterative processes (such as increasing or decreasing the transmit power level of a DMD15 modulator), the FSN is incremented after each message packet. When the FSN value reaches 255, it overflows and begins again at zero. The FSN tag is a powerful tool that assures sequential information framing, and is especially useful where commands require more than one message packet.

The full handshake/acknowledgment involves a reversal of source and destination ID codes in the next message frame, followed by a response code in the <OPCODE> field of the message packet from the equipment under control.

If a command packet is sent and not received at its intended destination, a timeout condition can occur because a response message is not received by the packet originator. On receiving devices slaved to an M & C computer, the timeout delay parameters may be programmed into the equipment in accordance with site requirements by Radyne Corp. prior to shipment, or altered by qualified personnel. The FSN handshake routines must account for timeout delays and be able to introduce them as well.

B.1.4 Global Response Operational Codes

In acknowledgment (response) packets, the operational code <OPCODE> field of the message packet is set to 0 by the receiving devices when the message intended for the device is evaluated as valid. The device that receives the valid message then exchanges the <SOURCE ID> with the <DESTINATION ID>, sets the <OPCODE> to zero in order to indicate that a good message was received, and returns the packet to the originator. This "GOOD MESSAGE" opcode is one of nine global responses. Global response opcodes are common responses, issued to the M&C computer or to another device, that can originate from and are interpreted by all Radyne equipment in the same manner. These are summarized as follows (all opcode values are expressed in decimal form):

RESPONSE OPCODE DESCRIPTION	OPCODE
Good Message	000
Bad Parameter	255
Bad Opcode	254
Bad Checksum	253
Command Not Allowed in LOCAL Mode	252
Command Not Allowed in AUTO Mode	251
Bad Destination	250
Unable to Process Command	249
Packet Too Long	248

Table 2. Response OPCODES

B.1.5 Software Compatibility

******CAUTION******

The DMD15 RLLP is not software-compatible with the following previous Radyne products: RCU5000 and DMD4500. These products may not occupy the same bus while using this protocol as equipment malfunction and loss of data may occur.

The COMMSPEC, operating in conjunction within the RLLP shell, provides for full forward and backward software compatibility independent of the software version in use. New features are appended to the end of the DATA field without OPCODE changes. Older software simply discards the data as extraneous information without functional impairment for backward compatibility.

If new device-resident or M&C software receives a message related to an old software version, new information and processes are not damaged or affected by the omission of data.

The implementation of forward and backward software compatibility often, but not always, requires the addition of new Opcodes. Each new function requires a new Opcode assignment if forward and backward compatibility cannot be attained by other means.

B.1.6 RLLP Summary

The RLLP is a simple send-and-wait protocol that automatically re-transmits a packet whenever an error is detected, or when an acknowledgment (response) packet is absent.

During transmission, the protocol wrapper surrounds the actual data to form information packets. Each transmitted packet is subject to time out and frame sequence control parameters, after which the packet sender waits for the receiver to convey its response. Once a receiver verifies that a packet sent to it is in the correct sequence relative to the previously received packet, it computes a local checksum on all information within the packet excluding the <SYN> character and the <CHECKSUM> fields. If this checksum matches the packet <CHECKSUM>, the receiver processes the packet and responds to the packet sender with a valid response (acknowledgment) packet. If the checksum values do not match, the receiver replies with a negative acknowledgment (NAK) in its response frame.

The response packet is therefore either an acknowledgment that the message was received correctly, or some form of a packetized NAK frame. If the sender receives a valid acknowledgment (response) packet from the receiver, the <FSN> increments and the next packet is transmitted as required by the sender.

However, if a NAK response packet is returned the sender re-transmits the original information packet with the same embedded <FSN>.

If an acknowledgment (response) packet or a NAK packet is lost, corrupted, or not issued due to an error and is thereby not returned to the sender, the sender re-transmits the original information packet; but with the same <FSN>. When the intended receiver detects a duplicate packet, the packet is acknowledged with a response packet and internally discarded to preclude undesired repetitive executions. If the M&C computer sends a command packet and the corresponding response packet is lost due to a system or internal error, the computer times out and re-transmits the same command packet with the same <FSN> to the same receiver and waits once again for an acknowledgment or a NAK packet.

To reiterate, the format of the message block is shown in below in Table 4, Link Level Protocol Message Block.

SYNC	COUNT	SRC ADDR	DEST ADDR	FSN	OP CODE	DATA BYTES	CHECKSUM

Table 4. Link Level Protocol Message Block

B.2 Remote Port Packet Structure:

The RCS11 protocol is an enhancement on the DMD5000 protocol. It also uses a packet structure format. The structure is as follows:

<SYNC> = Message format header character that defines the beginning of a message. The <SYNC> character value is always 0x16. (1 byte)

<BYTE COUNT> = Number of bytes in the <DATA> field. (2 bytes)

<SOURCE ID> = Identifies the address of the equipment from where the message originated. (1 byte)

<DEST. ID> = Identifies the address of the equipment where the message is to be sent. (1 byte)

<F.S.N.> = Frame sequence number insures correct packet acknowledgment and data transfers. (1 byte)

<OPCODE> = This byte identifies the message type associated with the information data. The equipment processes the data according to the value in this field. Return error codes and acknowledgment are also included in this field. (2 bytes)

<...DATA...> = Information data. The number of data bytes in this field is indicated by the <BYTE COUNT> value.

<CHECKSUM> = The modulo 256 sum of all preceding message bytes excluding the <SYNC> character. (1 byte)

******CAUTION******

The DMD15 RLLP is not software-compatible with the following previous Radyne products: RCU5000 and DMD4500. These products may not occupy the same bus while using this protocol as equipment malfunction and loss of data may occur.

NOTE: When transmitting a packet at 9600 baud, the Remote M&C should ensure that the timeout value between characters does not exceed the time it takes to transmit 200 characters (≈ 200 msec). If this timeout value is exceeded, the equipment will timeout.

RCS11 Opcode Descriptions

Regular Commands

Opcode <2000h> Query RCS11 Device Identification

Query Response Data Field (1 byte):

<1>	Device Identification	26 decimal for RCS11

Opcode <2080h> Query RCS11 Device Firmware Version

Query Response Data Field (19 bytes):

<1>	Device Firmware Version MSB	binary number, implied decimal point
<1>	Device Firmware Version LSB	binary number, implied decimal point
<10>	Device Firmware Name	null terminated string
<7>	Device Firmware Release Date	null terminated string

Opcode <2280h> Command RCS11 Load Factory Defaults

Command Data Field (2 bytes):

<1>	Unlock Code 1	0x55
<1>	Unlock Code 2	0xAA

Opcode <2083h> Query RCS11 Backup Configuration

Query Response Data Field (1 byte):

<1>	Backup Configuration	0 = Modem (Coupled)
		1 = Mod Only
		2 = Demod Only
		3 = Mod/Demod (Uncoupled)

Opcode <2084h> Query RCS11 Backup Mode

Query Response Data Field (1 byte):

<1>	Backup Mode	0 = Manual
		1 = Automatic
		2 = <unknown>

Opcode <2284h> Command RCS11 Backup Mode

Command Data Field (1 byte):

<1>	Backup Mode	0 = Manual
		1 = Automatic

Opcode <2085h> Query RCS11 Backup State

Query Response Data Field (2 bytes):

<1>	Mod Backup state	0 = A online
		1 = B online
		2 = <none or unknown>
<1>	Demod Backup State	0 = A online
		1 = B Online
		2 = <none or unknown>

Opcode <2285h> Command RCS11 Force Manual Backup

Command Data Field (2 bytes):

<1>	Mod Backup State	0 = A online
		1 = B online
<1>	Demod Backup State	0 = A online
		1 = B online

Opcode <2086h> Query RCS11 Switch Delays

Query Response Data Field (8 Bytes):

<1>	Switch Delay Mod Fault MSB
<1>	Switch Delay Mod Fault LSB
<1>	Switch Delay Mod NoFault MSB
<1>	Switch Delay Mod NoFault LSB
<1>	Switch Delay Demod Fault MSB
<1>	Switch Delay Demod Fault LSB
<1>	Switch Delay Demod NoFault MSB
<1>	Switch Delay Demod NoFault LSB

Opcode <2286h> Command RCS11 Switch Delays

Command Data Field (8 Bytes):

<1>	Switch Delay Mod Fault MSB
<1>	Switch Delay Mod Fault LSB
<1>	Switch Delay Mod NoFault MSB
<1>	Switch Delay Mod NoFault LSB
<1>	Switch Delay Demod Fault MSB
<1>	Switch Delay Demod Fault LSB
<1>	Switch Delay Demod NoFault MSB
<1>	Switch Delay Demod NoFault LSB

Opcode <2087h> Query RCS11 Switch Alarms

Query Response Data Field (5 bytes):

<1>	Global Current Alarm	Bit 0: 1 = Global Alarm Bit 1..7: unused, return 0
<1>	External Current Alarms	Bit 0: 1 = Mod A Alarm Bit 1: 1 = Force Mod A Alarm Bit 2: 1 = Demod A Alarm Bit 3: 1 = Force Demod A Alarm Bit 4: 1 = Mod B Alarm Bit 5: 1 = Force Mod B Alarm Bit 6: 1 = Demod B Alarm Bit 7: 1 = Force Demod B Alarm
<1>	Switch Current Alarms	Bit 0: 1 = Switch Power 1 Alarm Bit 1: 1 = Switch Power 2 Alarm Bit 2: 1 = Switch Firmware Alarm Bit 3: 1 = Switch NV Alarm Bit 4: 1 = Switch Internal Alarm Bit 5..7: unused, return 0
<2>	Spare Current Alarms	Always 0

Opcode: <2089h> Query RCS11 Switch Alarms Mask

Query Response Data Field (4 Bytes):

<1>	External Alarm Masks	1 = Enabled, 0 = Masked Off Bit 0: 1 = Mod A Alarm Enabled Bit 1: 1 = Force Mod A Alarm Enabled Bit 2: 1 = Demod A Alarm Enabled Bit 3: 1 = Force Demod A Alarm Enabled Bit 4: 1 = Mod B Alarm Enabled Bit 5: 1 = Force Mod B Alarm Enabled Bit 6: 1 = Demod B Alarm Enabled

		Bit 7: 1 = Force Demod B Alarm Enabled
<1>	Switch Alarm Masks	1 = Enabled, 0 = Masked Off Bit 0: 1 = Switch Power 1 Alarm Enabled Bit 1: 1 = Switch Power 2 Alarm Enabled Bit 2: 1 = Switch Firmware Error Enabled Bit 3: 1 = Switch NV Alarm Bit 4: 1 = Switch Internal Alarm Enabled Bit 5..7: unused, return 0
<2>	Spare Current Alarms	Always 0

Opcode: <2289h> Command RCS11 Switch Alarms Masks

Command Data Field (4 Bytes):

<1>	External Alarm Masks	1 = Enabled, 0 = Masked Off Bit 0: 1 = Mod A Alarm Enabled Bit 1: 1 = Force Mod A Alarm Enabled Bit 2: 1 = Demod A Alarm Enabled Bit 3: 1 = Force Demod A Alarm Enabled Bit 4: 1 = Mod B Alarm Enabled Bit 5: 1 = Force Mod B Alarm Enabled Bit 6: 1 = Demod B Alarm Enabled Bit 7: 1 = Force Demod B Alarm Enabled
<1>	Switch Alarm Masks	1 = Enabled, 0 = Masked Off Bit 0: 1 = Switch Power 1 Alarm Enabled Bit 1: 1 = Switch Power 2 Alarm Enabled Bit 2: 1 = Switch Firmware Error Enabled Bit 3: 1 = Switch NV Alarm Enabled Bit 4: 1 = Switch Internal Alarm Enabled Bit 5..7: unused, 0 filled
<2>	Spare Current Alarms	0 filled