



**TRICOM**  
RESEARCH, INC.

## **MVSS Controller Interface Control Document**



**90400-01267 Rev. P3**

# Table of Contents

<b>Table of Contents .....</b>	<b>ii</b>
<b>Revision History .....</b>	<b>iii</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose .....	1
<b>2. Interfaces .....</b>	<b>2</b>
2.1 Forward.....	2
2.2 Interface description.....	2
2.3 Interface message types.....	2
2.4 Requested status messages.....	2-6
2.5 Unprompted status messages.....	6

## **Revision History**

<b>Date</b>	<b>Description</b>	<b>Version</b>
09/01/16	Preliminary Release	P1
10/07/16	Update per firmware V1.1.4	P2
2/15/17	Update section 2.2	P3

## **1. Introduction**

**1.1 Purpose** – The purpose of this ICD is to describe the interface between the MVSS controller logic and the serial and Ethernet external control ports.

## 2. Interfaces

**2.1 Forward** – The MVSS Controller is a subsystem of the MVSS Radio responsible for interfacing to an external terminal for monitoring of the system status and to read and write to external general-purpose inputs and outputs. Additionally, the interface is used to configure system parameters, such as IP addresses, alarm thresholds and to load system firmware updates. Note that firmware updates are supported only through the serial interface. A separate GUI will be developed for modifying certain system variables not addressed in this document. The MVSS Radio Programming Document will govern the protocol and format for these settings and variables and for system firmware updates and this information is not a part of this document.

**2.2 Interface Description** – There are two interfaces to the MVSS Controller. They support a similar command structure – there may be a division of which commands are permitted on which interface in a future update. The first interface is through an RS-232 serial port to a typical terminal emulator such as PuTTY. Selecting Connection Type is Serial and settings Speed/bit parity are 9600 8,n,1. The second interface is through an Ethernet port where a RAW TCP session is established. Using PuTTY a raw session can be established by entering the IP address of the MVSS Controller, port number 10001 and selecting Connection Type: Raw.

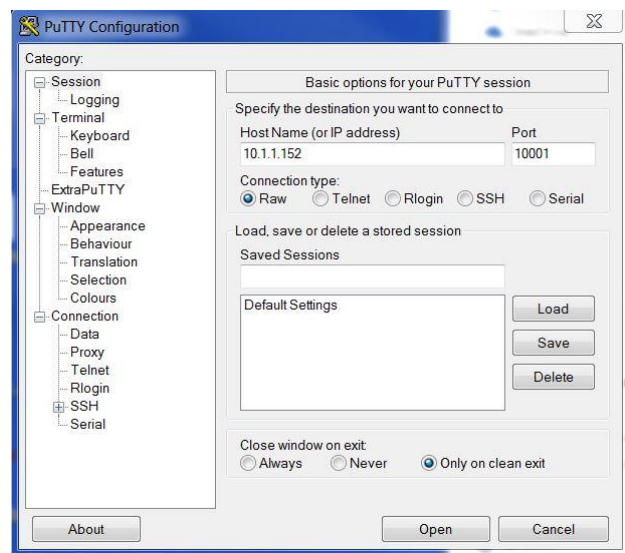
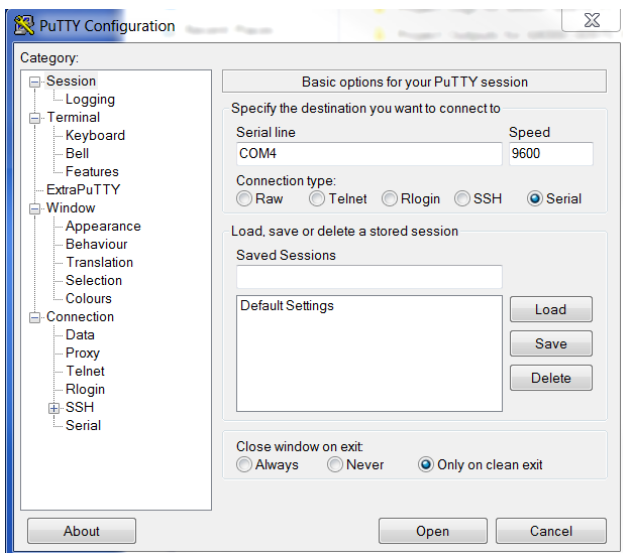


Figure 2.2.1 Serial Port PuTTY terminal emulator      Figure 2.2.2 RAW PuTTY terminal emulator

**2.3 Interface Message Types** – There are three types of messages between a remote terminal and the MVSS Controller. Square brackets are used to indicate the text within a request and the responses are shown in italic for readability of this document.

**2.4 Requested status messages** – These messages are responses from the MVSS Controller to a request from the remote terminal. A request is a character or string of characters followed by a carriage return <cr>. Note that there are two “batch” requests; [?] and [status] that return multiple responses. The [status] request is only supported on the Ethernet port. The requested status messages and responses are listed below:

### 2.4.1 [?] Request

[?] This is a request for a listing of all available commands on the current interface

Response (this is the current listing of the existing firmware revision)

```
close contact 1
close contact 2
close contact 3
close contact 4
open contact 1
open contact 2
open contact 3
open contact 4
read inputs
set controller ip www.xxx.yyy.zzz
set radio ip www.xxx.yyy.zzz
set virtual ip www.xxx.yyy.zzz
show controller ip
show radio ip
show virtual ip
eeread 0x<address> (note that this command reads directly from the eeprom)
eewrite 0x<address> 0x<data> (note that this command writes directly to the eeprom)
<ack>
```

### 2.4.2 [close contact] Request

[close contact n] where n is 1-4 closes the associated GPIO relay contact

Response

```
<ack>
```

### 2.4.3 [open contact] Request

[open contact n] where n is 1-4 opens the associated GPIO relay contact

Response

```
<ack>
```

### 2.4.4 [read inputs] Request

[read inputs] returns the state of all 4 GPIO inputs

Response

```
Input 1 = state  where state is open or closed
Input 2 = state  where state is open or closed
```

*Input 3 = state* where state is open or closed

*Input 4 = state* where state is open or closed

*<ack>*

#### **2.4.5 [set controller ip] Request**

[set controller ip n] where n is a valid ip address in the format www.xxx.yyy.zzz

Response

*<ack>*

#### **2.4.6 [set radio ip] Request**

[set radio ip n] where n is a valid ip address in the format www.xxx.yyy.zzz

Response

*<ack>*

#### **2.4.7 [set virtual ip] Request**

[set virtual ip n] where n is a valid ip address in the format www.xxx.yyy.zzz

Response

*<ack>*

#### **2.4.8 [show controller ip] Request**

[show controller ip]

Response

www.xxx.yyy.zzz

*<ack>*

#### **2.4.9 [show radio ip] Request**

[show radio ip]

Response

www.xxx.yyy.zzz

*<ack>*

#### **2.4.9.1 [show virtual ip] Request**

[show virtual ip]

Response

www.xxx.yyy.zzz

<ack>

### 2.4.9.1 [eeread] Request

[eeread 0xaa]            where aa is the address of the eeprom location

Response

0xbb                    where bb is the data in eeprom address location aa

<ack>

### 2.4.9.2 [eewrite] Request

[eewrite 0xaa 0xbb]    Where aa is the address and bb is the data

Response

<ack>

Note: The eeread and eewrite commands should not be executed except at the factory. If these commands are improperly used the system may be permanently corrupted requiring a new factory firmware load.

### 2.4.9.3 [status] Request

[status]

Response

*Primary Radio Battery Voltage* = n V            where n is the voltage  
*External Radio Battery Voltage* = n V            where n is the voltage  
*Radio Battery in Use* = source                    where source is either EXT DC or Primary  
*Aux Switch Primary Battery Voltage* = n V        where n is the voltage\*  
*Aux Switch External Battery Voltage* = n V        where n is the voltage\*  
*Aux Switch Battery in Use* = source                where source is either EXT DC or Primary\*  
*System Temperature* = n DEG C                    where n is the temperature  
*Aux Connector Sense Pins* = B1 n B2 n B3 n        where n is either 0 or 1  
*POE* = state    where state is either Enabled or Disabled  
*Input 1* = state                                        where state is either Open or Closed  
*Input 2* = state                                        where state is either Open or Closed  
*Input 3* = state                                        where state is either Open or Closed  
*Input 4* = state                                        where state is either Open or Closed  
*Output 1* = state                                      where state is either Open or Closed  
*Output 2* = state                                      where state is either Open or Closed  
*Output 3* = state                                      where state is either Open or Closed

**Tricom Research, Inc.**

<http://www.tricomresearch.com>

17981 Sky Park Circle, Suite M, Irvine, CA 92614

Ph: (949) 250-6024 fax: (949) 250-6023



Output 4 = state where state is either Open or Closed  
 Controller IP Address = www.xxx.yyy.zzz Where www.xxx.yyy.zz is the ip address  
 Radio IP Address = www.xxx.yyy.zzz Where www.xxx.yyy.zz is the ip address  
 Virtual IP Address = www.xxx.yyy.zzz Where www.xxx.yyy.zz is the ip address  
 <ack>

\* Note that these messages are only displayed if an auxiliary switch is connected to the system

**2.5 Unprompted status messages** - These messages are generated by the MVSS Controller and are sent to the remote terminal without being prompted to do so. The unprompted status messages are listed below:

1. *LOW VOLTAGE ALARM - X* where X is either RADIO EXTERNAL or RADIO PRIMARY
2. *LOW VOLTAGE ALARM CLEAR - X* where X is either RADIO EXTERNAL or RADIO PRIMARY
3. *LOW VOLTAGE SHUTDOWN*
4. *LOW VOLTAGE ALARM - X* where X is either AUX SWITCH PRIMARY or AUX SWITCH EXTERNAL
5. *LOW VOLTAGE ALARM CLEAR - X* where X is either AUX SWITCH PRIMARY or AUX SWITCH EXTERNAL
6. *HIGH TEMPERATURE ALARM* If high temperature alarm limit is reached
7. *HIGH TEMPERATURE ALARM CLEAR*
8. *HIGH TEMPERATURE SHUTDOWN* If high temperature shut down limit is reached – system then shuts itself off
9. *Input n status changed to state* where n is the GPIO number 1-4 and state is open or closed
10. *MANUAL SHUTDOWN SEQUENCE* When a power down sequence is initiated