High Level Design Document For

CMS - Wrapper Integration

Version: 1.0

Date: 09-Sept-2010

INDEX

Revision History

Version	Summary of Change	Written By	Approver	Date
1.0	Draft	Swati Jadhav	Nilesh Joshi	09/09/2010

1. Introduction

1.1 Purpose

This document specifies the High Level design for integration of the CMS and Hardware wrapper component. NCRA may include the wrapper design in this document.

1.2 Scope

The CMS would consist of a GUI interface from where users can invoke commands on the telescope control system based on pre-defined Roles/Privileges. The command would then be translated to an XML file and would be written to pre-defined folder allocated to a subsystem. Each Telescope subsystem would have its own wrapper and would read and write xml files to a pre-defined folder.
This document covers the details of integration of the CMS with the Hardware Subsystem wrapper. The subsystem considered here is as below:

- Servo Subsystem [Radio Telescope]

< NCRA may include the wrapper details for the other subsystems >

This document will cover the details of request/response formats between CMS and above mentioned Subsystems.

1.3 Glossary & Acronyms

IUCAA – Inter-University Centre for Astronomy and Astrophysics
NCRA –National Centre for Radio Astrophysics
URL – Uniform Resource Locator
I/O – Input/output
IDE – Integrated Development Environment
JDE – Java Development Environment
HTTP – Hypertext Transfer Protocol
J2EE – Java 2 Platform Enterprise Edition
JSP – Java Server Pages

1.4 References

Please refer the following documents available:

- Telescope_CMS_Requirement_Specifications_Document_V2[1].0.pdf
- command_structure_ver1.3.pdf
- TCS_Network_ICD_v2.4.pdf
- Servo_Telemetry_Commn_Protocol.pdf
- Technical Discussion meeting with IUCAA team on 26-Aug-2010
- Technical Discussion meeting with NCRA team on 03-Sept-2010

2. System Overview

The CMS consists of the following Components:

1. Web Interface (Presentation Layer)
2. Authentication Module
3. DateTimeLocation Provider Module
4. SubsytemXMLRequest/Response Handler
5. Servo Interface Module

The wrapper would consist of the following Components:

1. A XML Command parser
2. Translator module to translate the command to low level Hardware commands
3. An XML writer to write the response and status parameters
   < NCRA may add other component here>

3. Design Considerations

1.5 Assumptions and Dependencies

The following are the assumptions:

1. The execution of telescope control commands is dependent on the execution of the command by the wrapper component and the subsequent response generated.
2. If the response is not received within 1 min after sending the request to wrapper component, the request timeout message will be displayed to user.
   < NCRA team may modify the performance depending on the response for the command of the particular module >
3. A queuing mechanism would be used to generate command requests. Critical commands will have higher priority. It is assumed that the critical commands would be sent by Expert role user only.

1.6 Goals and Guidelines

For CMS:

- The User interface should be simple and easy to use.
- The error messages must be clear and concise.
- Should generate valid xml request as defined.
- Should update the UI within 3 seconds timeline, at least within the LAN.

For Wrapper:

- Should generate valid xml response as defined.
- Should return response within the stipulated time.
  < NCRA team needs to include the details >

1.7 Development Methods

For CMS:

CMS web application development tools:

1. Eclipse(Europa) IDE 3.3
2. Java EE 5.0
3. Tomcat 6 (supported on JDK 1.6)
4. MySQL server v1.5
5. JQuery JavaScript Library (this will be enclosed in CMS web application)
6. Hibernate v3.2.6 Library (this will be enclosed in CMS web application)

For Wrapper:

< NCRA team needs to include the details >

4. System Architecture

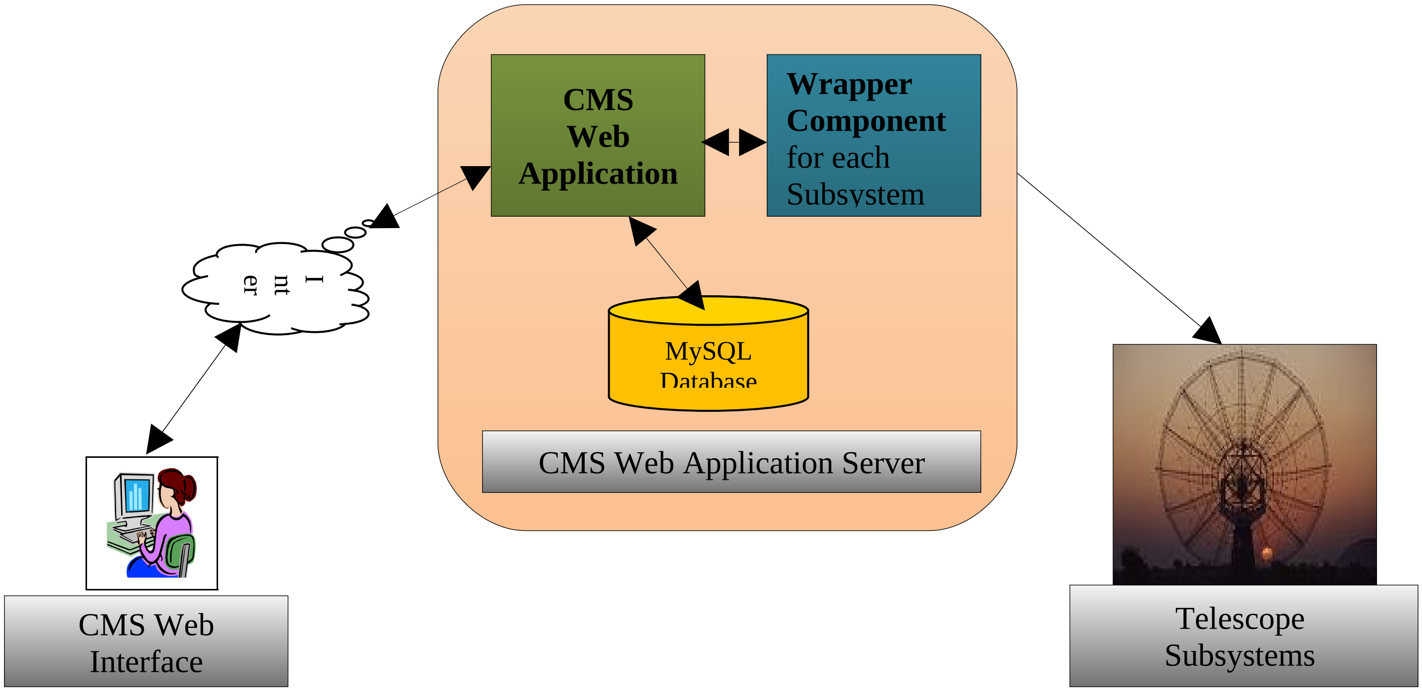

Figure 1 CMS Application System Architecture

The System Architecture consists of following components:

1. CMS Web Interface
2. CMS Web Application
3. Wrapper Component
4. MySQL Database
5. Telescope Subsystems

The Application flow for Expert Level Command is as follows:

1. User selects the command from a drop down.
2. Once selected the command syntax and a command sample is available on screen
3. User clicks on execute after entering the proper command data.
4. The request is sent to Subsystem XML Request/Response Handler applicable for that Subsystem to which the command is to be sent
5. The Subsystem XML Request/Response Handler module creates and XML request file and places the same in the pre-defined folder.
6. The wrapper component associated with the Subsystem, which is continuously monitoring for a request picks up the request file and deletes it after reading.

7. The wrapper component then parses the xml to get the command data and converts into the Hardware understandable command.
8. Once the wrapper component sends the command to the Hardware component, it will get the response from the Hardware component.
9. On receiving the response the wrapper component creates a response xml with the received response in a pre-defined folder.
10. The Subsystem XML Request/Response Handler on sending request starts monitoring for the response xml.
11. On receiving the response xml, it decodes the response and updates the GUI with appropriate Message/Data.
12. In case of error or request time out, appropriate message is displayed to the user.
13. The Logging of request sent and response received would be done as when the request and response are generated.

5. System Requirements

For CMS:

1. An application server that can host a J2EE 1.5 application, to be specific the application server should support at least Servlet 2.4 and JSP 2.0 specifications. For e.g. Apache Geronimo 3.0 server.
2. The application server and the load balancer should be able to handle the sessions. "Sticky Sessions" with cookie persistence or any other session management technique should be supported by the load balancer.
3. MySQL 5.0

For Wrapper (Individual Subsystems Details)

< NCRA team needs to include the details >

6. Detailed System Design

1.8 CMS Web Application

The CMS Web Application is a J2EE 1.5 compliant application hosted on Apache Geronimo server or on Tomcat 5.5 server in Development Environment. It consists of the following components:

- UI/Presentation layer

The presentation layer would consist of JSP pages which would allow the user to do the following through Expert Tab:

  - Select a Command
  - Execute the selected command
  - View the command execution status

- Authentication Module

  This module will authenticate the user. The user would enter his/her username and password from Login page. The user credentials would be authenticated with database values. Once the credentials are authenticated the user is displayed the Web UI as per his Role/Privilege.

- DateTimeLocation Provider Module

  The module provides the Date, Time (LST, IST, UTC) to the Web UI.

- Subsystem XML Request/Response Handler

  This module handles the request and response for a particular system. It generates the request XML on the basis of the Command Syntax received from particular subsystem. Then it monitors for the response XML. On receiving the response XML it will be parse the response and send the message to the UI Layer.

- Servo Interface Module
  This module will handle the commands related to Servo module. This would involve the following:
  - Loading the Servo commands on the GUI Command List (Expert Tab).
  - Validation of the Servo command entered by user.
  - Generating the required request format.
  - Monitoring for the response xml
  - Validating the response XML against standard response format
  - Updating the UI with the appropriate response

1.9 Subsystem Wrapper Application

Each Telescope subsystem would have its own wrapper component. Each wrapper component would consist of following components:

- XML Request/Response Handler
  This module will wait for incoming requests. Once a request is received it will send it to XML parser to parse the request. Based on the request the Low level Hardware command would be formed by the Hardware Interface module and would sent to the corresponding Subsystem Hardware layer. The response

received from the Hardware layer would be converted into XML format by this module and would be written to a pre-defined folder.

- A XML Command parser
  This module will parse the request received
- Translator module to translate the command to low level Hardware commands
- An XML writer to write the response and status parameters

< NCRA team needs to include the details for other Subsystem specific modules >

7. CMS-Wrapper Communication

1.10 Servo Subsystem

All request command will be enclosed with `<COMMAND> </COMMAND>` tag. Each response command will be enclosed with `<RESPONSE> </RESPONSE>` tag.
The request and response file will have all their respective commands and responses enclosed in `<SERVO_Module> </SERVO_Module>` tag.
The commands would be sent in SERVO_Request.xml file. A sample request would be as below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<SERVO_Module>
      <TIMESTAMP>1283928575000</TIMESTAMP>
      <COMMAND>
            <ID>POSITION</ID>
            <DATA>POSITION ax ang1 ang2</DATA>
      </COMMAND>
</SERVO_Module>
```

The response would be received in SERVO_Response.xml file. A sample response would be as below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<SERVO_Module>
      <TIMESTAMP>1283928575000</TIMESTAMP>
      <RESPONSE>
            <ID>POSITION</ID>
            <ACK>
                  <CODE>1</CODE>
                  <MESSAGE>Detailed Error/status Msg</MESSAGE>
            </ACK>
      <RESPONSE>
</SERVO_Module>
```

- **General request format:**

```
<COMMAND>
      <ID>[command name]</ID>
      <DATA>[command data]</DATA>
</COMMAND>
```

- **General response format:**

Intermediate Response:
ACCEPTED:

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>10</CODE>
      </ACK>
</RESPONSE>
```

NOT ACCEPTED:

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>11</CODE>
      </ACK>
</RESPONSE>
```

CMD TIMEOUT:

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>11</CODE>
            <MSG>CMD TIMEOUT</MSG>
      </ACK>
</RESPONSE>
```

SYNTAX ERROR:

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>11</CODE>
            <MSG>SYNTAX ERRO</MSG>
      </ACK>
</RESPONSE>
```

ILLEGAL CMD:

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>11</CODE>
            <MSG>ILLEGAL CMD</MSG>
      </ACK>
</RESPONSE>
```

Note: <MSG> tag is optional; it can be used to send additional error details

Success Response:
SUCCESSFUL:

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>10</CODE>
      </ACK>
</RESPONSE>
```

Failure Responses:
IRRELEVAL COMMAND:

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>255</CODE>
      </ACK>
</RESPONSE>
```

FAILED:

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>20</CODE>
      </ACK>
</RESPONSE>
```

ABORTED:

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>30</CODE>
      </ACK>
</RESPONSE>
```

EVENT Responses:

NOT STOWING AZ:

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>62</EVENT>
</RESPONSE>
```

NOT STOWING EL

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>63</EVENT>
</RESPONSE>
```

NOT STOW RELEASING AZ

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>64</EVENT>
</RESPONSE>
```

NOT STOW RELEASING EL

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>65</EVENT>
</RESPONSE>
```

CAN’T SWITCH ON AXIS AZ

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
```

```
        </ACK>
        <EVENT>66</EVENT>
    </RESPONSE>
```

CAN’T SWITCH ON AXIS EL

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>67</EVENT>
    </RESPONSE>
```

AXIS NOT TURNING ON AZ

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>68</EVENT>
    </RESPONSE>
```

AXIS NOT TURNING ON EL

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>69</EVENT>
    </RESPONSE>
```

AXIS NOT TURNING OFF AZ

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>6a</EVENT>
    </RESPONSE>
```

AXIS NOT TURNING OFF EL

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
```

```
                  <CODE>12</CODE>
            </ACK>
            <EVENT>6b</EVENT>
      </RESPONSE>
```

STOW ERROR OCCURED AZ

```
      <RESPONSE>
            <ID>Command ID</ID>
            <ACK>
                  <CODE>12</CODE>
            </ACK>
            <EVENT>6c</EVENT>
      </RESPONSE>
```

STOW ERROR OCCURED EL

```
      <RESPONSE>
            <ID>Command ID</ID>
            <ACK>
                  <CODE>12</CODE>
            </ACK>
            <EVENT>6d</EVENT>
      </RESPONSE>
```

STOW NOT RELEASED AZ

```
      <RESPONSE>
            <ID>Command ID</ID>
            <ACK>
                  <CODE>12</CODE>
            </ACK>
            <EVENT>6e</EVENT>
      </RESPONSE>
```

STOW NOT RELEASED EL

```
      <RESPONSE>
            <ID>Command ID</ID>
            <ACK>
                  <CODE>12</CODE>
            </ACK>
            <EVENT>6f</EVENT>
      </RESPONSE>
```

NOT STOWED AZ

```
      <RESPONSE>
```

```
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>70</EVENT>
    </RESPONSE>
```

NOT STOWED EL

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>71</EVENT>
    </RESPONSE>
```

STOW RELEASING AZ

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>80</EVENT>
    </RESPONSE>
```

STOW RELEASING EL

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>81</EVENT>
    </RESPONSE>
```

STOW RELEASED AZ

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>82</EVENT>
    </RESPONSE>
```

STOW RELEASED EL

```
        <RESPONSE>
                <ID>Command ID</ID>
                <ACK>
                        <CODE>12</CODE>
                </ACK>
                <EVENT>83</EVENT>
        </RESPONSE>
```

BRAKE RELEASED AZ

```
        <RESPONSE>
                <ID>Command ID</ID>
                <ACK>
                        <CODE>12</CODE>
                </ACK>
                <EVENT>84</EVENT>
        </RESPONSE>
```

BRAKE RELEASED EL

```
        <RESPONSE>
                <ID>Command ID</ID>
                <ACK>
                        <CODE>12</CODE>
                </ACK>
                <EVENT>85</EVENT>
        </RESPONSE>
```

AXIS HELD AZ

```
        <RESPONSE>
                <ID>Command ID</ID>
                <ACK>
                        <CODE>12</CODE>
                </ACK>
                <EVENT>86</EVENT>
        </RESPONSE>
```

AXIS HELD EL

```
        <RESPONSE>
                <ID>Command ID</ID>
                <ACK>
                        <CODE>12</CODE>
                </ACK>
                <EVENT>87</EVENT>
        </RESPONSE>
```

STOWING AZ

```
        <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>88</EVENT>
    </RESPONSE>
```

STOWING EL

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>89</EVENT>
    </RESPONSE>
```

STOWED AZ

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>8a</EVENT>
    </RESPONSE>
```

STOWED EL

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>8b</EVENT>
    </RESPONSE>
```

BRAKED AZ

```
    <RESPONSE>
        <ID>Command ID</ID>
        <ACK>
            <CODE>12</CODE>
        </ACK>
        <EVENT>8c</EVENT>
    </RESPONSE>
```

BRAKED EL

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>8d</EVENT>
</RESPONSE>
```

POSITIONED AZ

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>8e</EVENT>
</RESPONSE>
```

POSITIONED EL

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>8f</EVENT>
</RESPONSE>
```

TRACKING AZ

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>90</EVENT>
</RESPONSE>
```

TRACKING EL

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>91</EVENT>
</RESPONSE>
```

PARKING AZ

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>92</EVENT>
</RESPONSE>
```

PARKING EL

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>93</EVENT>
</RESPONSE>
```

POSITIONING AZ

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>94</EVENT>
</RESPONSE>
```

POSITIONING EL

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>95</EVENT>
</RESPONSE>
```

MOTOR CURRENTS HIGH AZ

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>a0</EVENT>
</RESPONSE>
```

MOTOR CURRENTS HIGH EL

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>a1</EVENT>
</RESPONSE>
```

WIND VELOCITY HIGH

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>a2</EVENT>
</RESPONSE>
```

MOTOR INTERLOCK ERROR AZ

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>a3</EVENT>
</RESPONSE>
```

MOTOR INTERLOCK ERROR EL

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>a4</EVENT>
</RESPONSE>
```

EMERGENCY PARK STARTED

```
<RESPONSE>
      <ID>Command ID</ID>
      <ACK>
            <CODE>12</CODE>
      </ACK>
      <EVENT>a5</EVENT>
```

```
</RESPONSE>
```

- **Command Request/Response format:**

Operational Commands

1. Command 01: COLDSTART

Request format:

```
<COMMAND>
      <ID>COLDSTART</ID>
      <DATA>COLDSTART</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>COLDSTART</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

Where `<CODE>1</CODE>` indicates success.

2. Command 02: POSITION

Request format:

```
<COMMAND>
      <ID>POSITION</ID>
      <DATA>POSITION ax ang1 ang2…</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>POSITION</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

Where `<CODE>1</CODE>` indicates success.

3. Command 03: TRACK

Request format:

```
<COMMAND>
      <ID>TRACK</ID>
      <DATA>TRACK ax time ang1 ang2…</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>TRACK</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

Where `<CODE>1</CODE>` indicates success.

4. Command 04: HOLD

Request format:

```
<COMMAND>
      <ID>HOLD</ID>
      <DATA>HOLD ax</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>HOLD</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

Where `<CODE>1</CODE>` indicates success.

5. Command 05: STOP

Request format:

```
<COMMAND>
      <ID>STOP</ID>
      <DATA>STOP ax</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>STOP</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

Where `<CODE>1</CODE>` indicates success.

6. Command 06: CLOSE

Request format:

```
<COMMAND>
      <ID>CLOSE</ID>
      <DATA>CLOSE</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>CLOSE</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

Where `<CODE>1</CODE>` indicates success.

7. Command 07: STOW

Request format:

```
<COMMAND>
      <ID>STOW</ID>
      <DATA>STOW ax</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>STOW</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
```

```
</RESPONSE>
```

Where <CODE>1</CODE> indicates success.

8. Command 08: STOW RELEASE

Request format:

```
<COMMAND>
      <ID>STOWRELEASE</ID>
      <DATA>STOWRELEASE ax</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>STOWRELEASE</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

Where <CODE>1</CODE> indicates success.

- **Display Commands**

9. Command 09: READ ANGLES

Request format:

```
<COMMAND>
      <ID>READANGLES</ID>
      <DATA>READANGLES</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>READANGLES</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
      <TIME>time</TIME>
      <AZCP>ang1</AZCP>
      <AZTP>ang1</AZTP>
      <AZPP>ang1</AZPP>
      <ELCP>ang1</ELCP>
      <ELTP>ang1</ELTP>
      <ELPP>ang1</ELPP>
<RESPONSE>
```

Where <CODE>1</CODE> indicates success,
<TIME>time</TIME> specifies the time of day
<AZCP>ang1</AZCP> specifies az current position (CP)
<AZTP>ang1</AZTP> specifies az target position (TP)
<AZPP>ang1</AZPP> specifies az pot position (PP)
<ELCP>ang1</ELCP> specifies el current position (CP)
<ELTP>ang1</ELTP> specifies el target position (TP)
<ELPP>ang1</ELPP> specifies el pot position (PP)

10. Command 10: READ ANALOG VARS

Request format:

```
<COMMAND>
      <ID>READANALOGVARS</ID>
      <DATA>READANALOGVARS</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>READANALOGVARS</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
      <TIME>time</TIME>
      <AZM1C>xxx.xxxx</AZM1C>
      <AZM2C>xxx.xxxx</AZM2C>
      <AZT1>xxxx.xxxx</AZT1>
      <AZT2>xxxx.xxxx</AZT2>
      <ELM1C>xxx.xxxx</ELZM1C>
      <ELM2C>xxx.xxxx</ELM2C>
      <ELT1> xxxx.xxxx</ELT1>
      <ELT2> xxxx.xxxx</ELT2>
      <WINDVEL1>windvel</WINDVEL1>
      <WINDVEL2>windvel</WINDVEL2>
</RESPONSE>
```

Where <CODE>1</CODE> indicates success,
<TIME>time</TIME> specifies the time of day
<AZM1C>xxx.xxxx</AZM1C> specifies az motor1 current in amps
<AZM2C>xxx.xxxx</AZM2C> specifies az motor2 current in amps

`<AZT1> xxxx.xxxx</AZT1>` specifies az tacho1 in RPM
`<AZT2> xxxx.xxxx</AZT2>` specifies az tacho2 in RPM
`<ELM1C>xxx.xxxx</ELZM1C>` specifies el motor1 current in amps
`<ELM2C>xxx.xxxx</ELM2C>` specifies el motor2 current in amps
`<ELT1> xxxx.xxxx</ELT1>` specifies el tacho1 in RPM
`<ELT2> xxxx.xxxx</ELT2>` specifies el tacho2 in RPM
`<WINDVEL1>windvel</WINDVEL1>` specifies wind vel in KMPH
`<WINDVEL2>windvel</WINDVEL2>` specifies wind vel in KMPH

11. Command 11: READ DIGITAL VARS

Request format:

```
<COMMAND>
      <ID>READDIGITALVARS</ID>
      <DATA>READDIGITALVARS</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID> READDIGITALVARS</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
      <TIME>time</TIME>
      <BYTE0>35H</BYTE0>
      <BYTE1>data</BYTE1>
      <BYTE2>data</BYTE2>
      <BYTE3>data</BYTE3>
      <BYTE4>data</BYTE4>
      <BYTE5>data</BYTE5>
      <BYTE6>data</BYTE6>
<RESPONSE>
```

Where `<CODE>1</CODE>` indicates success,
`<TIME>time</TIME>` specifies the time of day
`<BYTE0>35H</BYTE0>` specifies the digital data response code
`BYTE1` to `BYTE6` specify status of the various digital variables.

12. Command 12: READ SET PARAMETERS

Request format:

```
<COMMAND>
```

```
      <ID>READSETPARAMETERS</ID>
      <DATA> READSETPARAMETERS</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID> READSETPARAMETERS</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
      <RESPONSECODE>37H</RESPONSECODE>
      <TIME>time</TIME>
      <KP>xxx.xxxx</KP>
      <KI>xxx.xxxx</KI>
      <AZSOFTOFST>ang1</AZSOFTOFST>
      <AZSTOWANGLE>ang1</AZSTOWANGLE>
      <ELSOFTOFST>ang1</ELSOFTOFST>
      <ELSTOWANGLE>ang1</ELSTOWANGLE>
      <WINDVELLIMIT>windvel< /WINDVELLIMIT>
</RESPONSE>
```

Where `<CODE>1</CODE>` indicates success,

`< RESPONSECODE>37H</ RESPONSECODE>` specifies the set parameter response code

`<TIME>time</TIME>` specifies the time of day

`<KP>xxx.xxxx</KP>` specifies kp parameter

`<KI>xxx.xxxx</KI>` specifies ki parameter

`<AZSOFTOFST>ang1</AZSOFTOFST>` specifies az soft ofst

`<AZSTOWANGLE>ang1</AZSTOWANGLE>` specifies az stow angle

`<ELSOFTOFST>ang1</ELSOFTOFST>` specifies el soft ofst

`<ELSTOWANGLE>ang1</ELSTOWANGLE>` specifies el stow angle

`<WINDVELLIMIT>windvel< /WINDVELLIMIT>` specifies wind vel limit

- **Set Mode Commands**

13. Command 13: Set time of day

Request format:

```
<COMMAND>
      <ID>SETTIME</ID>
      <DATA>SETTIME time date</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>SETTIME</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

14. Command 14: Set stow angles

Request format:

```
<COMMAND>
      <ID>SETSTOWANGLE</ID>
      <DATA>SETSTOWANGLE ax ang1 ang2…</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID> SETSTOWANGLE</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

15. Command 15: Set S/W Hi Limit

Request format:

```
<COMMAND>
      <ID>SET_SW_HILIMIT</ID>
      <DATA>SET_SW_HILIMIT ax ang1 ang2…</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>SET_SW_HILIMIT</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

16. Command 16: Set S/W Lo Limit

Request format:

```
<COMMAND>
      <ID>SET_SW_LOLIMIT</ID>
      <DATA>SET_SW_LOLIMIT ax ang1 ang2…</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>SET_SW_LOLIMIT</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

17. Command 17: Set Windvel Limit

Request format:

```
<COMMAND>
      <ID>SET_WINDVEL</ID>
      <DATA>SET_WINDVEL windvel</DATA>
</COMMAND>
```

Response format:

```
<RESPONSE>
      <ID>SET_ WINDVEL</ID>
      <ACK>
            <CODE>1</CODE>
      </ACK>
</RESPONSE>
```

< NCRA team needs to include the details for other Subsystems >

8. Outstanding Issues

Issues	Responsibility