Plishner Radio Astronomy and Space Science Center
60-Foot Dish Position Indication System Development

David Molter — Deep Space Exploration Society

Glenn Davis — Deep Space Exploration Society

Richard Russel — Deep Space Exploration Society

Abstract

The Plishner Radio Astronomy and Space Science Center is operated by the Deep Space Exploration Society based out of Colorado Springs, Colorado. The largest antenna system is a 60- ft. parabolic reflector. This paper describes the development of a position indication system that enables precision pointing for radio astronomy experiments as well as Neutral Hydrogen detection (1420.406 MHZ) plus Earth-Moon-Earth and tropospheric communications at 1296 MHZ, 435 MHz and 144 MHz frequencies.

1.0 Introduction

The Deep Space Exploration Society (DSES) is a not for profit organization whose primary purposes are research and education of space communications and radio astronomy. The DSES primary asset is the Plishner Radio Astronomy and Space Science Center which is located in a radio quiet area near Haswell, Colorado.

1.1 60 Foot Radio Telescope

The Plishner Telescope (figure 1) was originally built by the National Bureau of Standard to study radio propagation of the ionosphere. It was built in the late 1950's and was transferred to the Air Force and the Army before it was declared as surplus in the 1970s. It was sold to Radio Research in the 1980s and was donated to DSES in 2009. The original group of DSES volunteers performed maintenance and refurbishment on the telescope to bring it back to operation. The original gear train for the antennas elevation and azimuth axis was retained. As part of the refurbishment a 3.5 KW 300-volt AC servomotor was installed in the azimuth drive to replace the original azimuth motor. A motor control chassis was built to handle the Azimuth Motor Controller and the Elevation Variable Frequency Drive units. A hand paddle was built to perform manual control of the azimuth and elevation circuitry.

Figure 1: DSES 60 ft. Dish

The site sat dormant for a couple of years with basic maintenance being performed. In the last two years, there has been resurgence in interest and many new people are joining DSES. When this current group became involved with the DSES group all of the antenna pointing was performed manually using the hand paddle.

The original elevation readouts were synchro rotating dials and were not operational since the resolvers mounted in the antenna had been removed and 12 bit absolute encoders were installed for the previous antenna position readout effort.

The limited financial resources (none) that were available required that the maximum amount of previously installed equipment be used to build a working azimuth and elevation readout circuit. The motor controller and Variable Frequency Drive (VFD) provided adequate power to run the motors in local control so a computer control would need to emulate the switch and potentiometer functions provided by the hand paddle. A spare encoder was available so the encoders installed in the mount would not have to be removed during the development and initial testing phase of the build.

This paper shows the design and performance analysis of the new digital pointing indicator system.

1.2 Multiband Feed System

The multi-band feed (1) was designed to be installed on the 60-foot diameter dish at the DSES site at Haswell, CO as seen in figure 2.

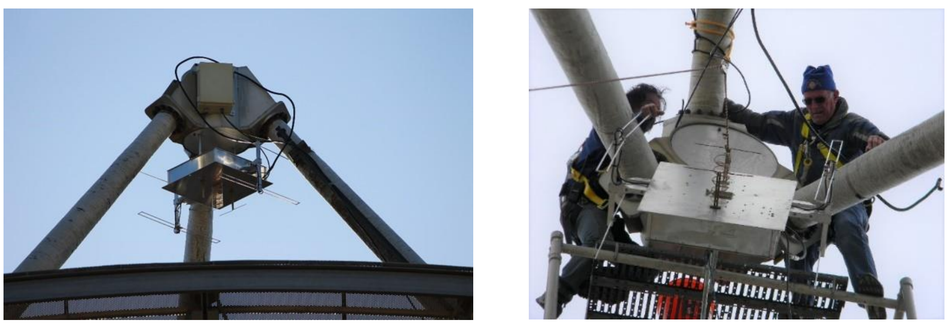


Figure 2: Multiband Feed and Installation Team

The feed controller (figure 3) provides direct switching of feed preamplifiers and provides power supplies.

Figure 3: Feed Controller

The radio signal is ported from the dish to a close communications trailer with equipment racks with receiving and analysis electronics (figure 4).

Figure 4: Communications Trailer and Equipment Racks

The 60-foot diameter parabolic dish is massively strong and has sat for years in a harsh environment. The surface finish root-mean-square (RMS) holds its shape up to full capability from 35 mph winds up to 50 mph winds. The dish is stored in the 90-degree elevation in order to limit any wear or tear due to wind load. In most cases, there are no spare parts. The broken parts must be removed and taken to a machinist and reproduced. The DSES team replaced the 3-phase drive motors through eddy current couplers with three phase variable frequency drive motors. The controller develops the third phase. The 3-phase power for the site was removed years before DSES processed the property. The site is now equipped with a single-phase propane powered generator.

The antenna position control can be operated locally or inside the communications van. The position indicator can be monitored next to the antenna controller. This allows the operator to control the antenna using azimuth-elevation or Right Ascension-Declination indications.

The dish is also used for the DSES club station (K0PRT) and will provide EME and tropospheric communications capabilities (figure 5).

Figure 5: DSES Club Station (K0PRT)

2.0 Position Indication System Development (Hardware)

The position indication system consisted of hardware and software development efforts that were integrated to provide the precision required for astronomical observations.

2.1 Hardware Overview

The hardware converts the mechanical positions of the antenna into a format that can be read by the computer. Figure 6 is a block diagram of the position indication system.

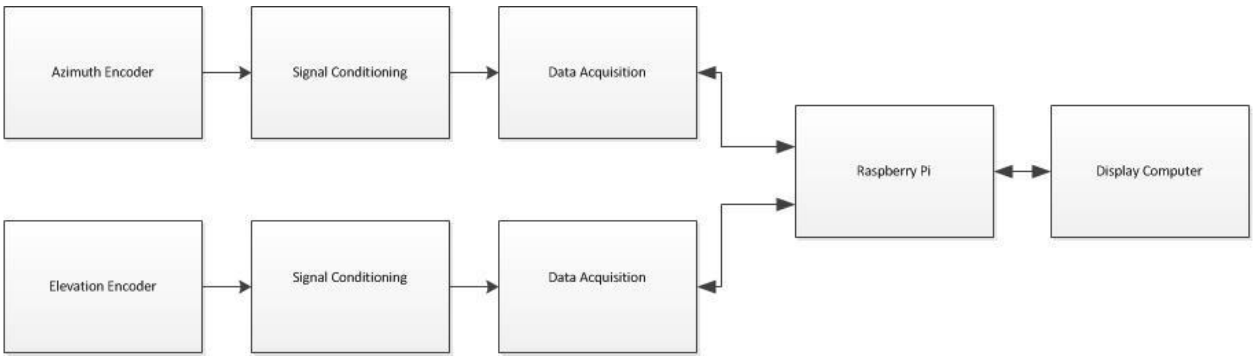


Figure 6: Position Indication System Circuitry Block Diagram

Figure 6 shows the signal path for the system:

- The antenna is on the left with the azimuth and elevation encoders mounted into the respective gear trains by flexible couplings.
- The 12 bits of parallel position data comes out of the absolute encoders (figure 7) as an open collector signal (signal is either open or grounded) into the signal conditioning board where a bank of parallel pull up resistors in series with LEDs not only change the input to a 0 VDC or 5 VDC signal suitable for input to the Digital Acquisition board, but also display if a bit is active in negative logic (lit LED means the bit is a 0).
- The 12 bits are then passed through the next board and are unchanged. This board will be used as an input for the mount limit switches and an output for relays to drive the motor controller or VFD depending upon which axis the board controls.
- The 12-bit signal is now at the input buffers of the digital acquisition board.
- The digital acquisition board takes the 12-bit digital signal and puts it into a RS-232 word that is output

2.2 Encoder Description

The Encoder is an Advanced Micro Controls Inc. model DC25F-B1A8BS. (2) This 12-bit absolute encoder (figure 7) has multiple configurations for the output signals to be formatted. The DSES setup uses 12 parallel output lines that are configured as open collector (each output line is either grounded when active or open when inactive). This encoder was installed by the previous group of volunteers and found to be adequate to get the site up to basic operation. Different models of encoders with higher resolution may be used in the future if a more accurate pointing dish is required.

Figure 7: 12-Bit Encoder

2.3 Signal Conditioner Description

The Signal Conditioner Board (figure 8) is a custom-built board that takes the 12 output lines of the encoder and converts the signals to a format that can be read by the Digital Acquisition Board. The board is composed of 12 parallel pullup resistors in series with LED indicators. Using pullup resistors converts the open or ground outputs of the encoder into +5V and 0V signal levels. A byproduct of this feature is that the signal level that is present on the line will be displayed by the lit LED as negative logic. (Lit LED shows that the signal line is grounded).

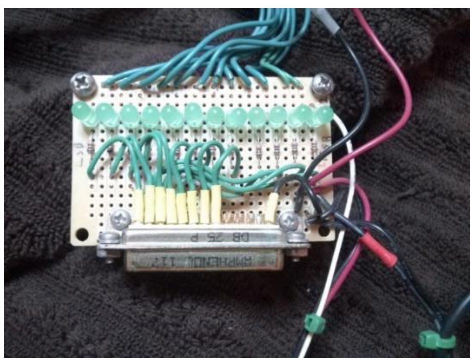


Figure 8: Signal conditioning Board

2.4 Digital Acquisition Board Description

The Digital Acquisition board (figure 9) is an Integrity Instruments model 232M3ADLT (3) . This model of acquisition board was selected because the engineer had previous experience with this unit. It was available with an extended temperature range that is required since the unit is used in a non-temperature controlled area of the dish. The acquisition board has 16 programmable digital input/output channels of which 12 are configured as the inputs from the encoder. The remaining 4 channels will be used in the future for limit switch inputs, motor controller enable and brake release lines. The acquisition board provides status and receives commands using a RS-232 interface. When the board is polled the data that is present in the buffer is read out and formatted to a serial word. The cable that connects the azimuth and elevation acquisition boards to the Raspberry Pi also converts the RS-232 serial to USB 2.0 serial. Each of the acquisition boards has a unique ID so they may be individually polled. The acquisition board also has (8) 12 bit analog inputs and (2) 12 bit analog outputs. These features may be used in the future.

Figure 9: Digital Acquisition Board

2.5 Raspberry Pi

The Raspberry Pi is a Pi 3 model B (figure 10). The Pi is used to read the serial USB 2.0 interface to the acquisition boards. This model was selected due to its having multiple USB ports, a small footprint and ease of programmability. There is a two line 16-character display that sits atop the Pi. The display shows the current azimuth and elevation values that is read off of the USB port. (4) and (5)

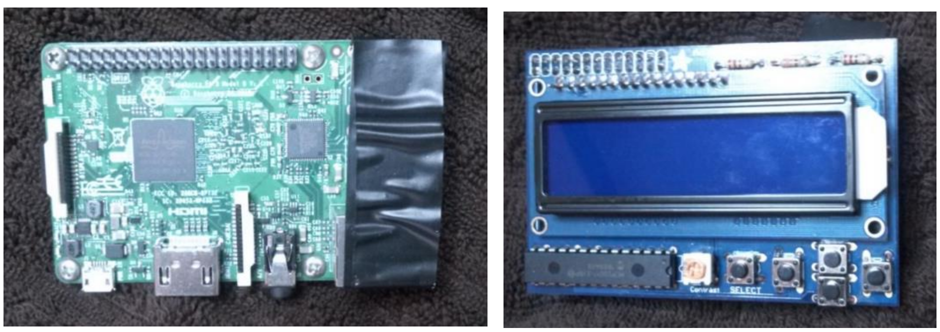


Figure 10: Rasberry Pi Computer and 2 Line 16 Character Display

2.6 Multi-Voltage Power Supply

The power supply (figure 11) has a standard 115 VAC input and provide 3 different output voltages. The model ECP40UT01 power supply (6) is a variable frequency switching power supply that provides 5 VDC at 5.0 A that is used by the Power Conditioning boards to power the pullup resistors and the Raspberry Pi Computer. The power supply provides +12 VDC at 2.0 A is used to power the Digital acquisition boards and will be used for future capabilities. -12 VDC at 0.5 A is also available and is provided for future capabilities.

Figure 11: Multi-Voltage Power Supply

2.7 Dish Manual Control Paddle

The Dish Manual Control Paddle (figure 12) is a box that is connected by a 60-foot cable into the unit that houses the azimuth motor controller and the elevation variable frequency drive. Control of each axis direction and speed is available and both axis can be operated at the same time. This type of a unit is normally used only for maintenance action, however at the moment this is the only way that we can move the dish.

Figure 12: Dish Manual Control Paddle

2.8 Azimuth Motor Controller and Elevation Variable Frequency Drive

Located on the right in the controller cabinet (figure 13) is the 1305 Adjustable Frequency AC Drive (7) provides power to the AC motor that is connected to the elevation gearbox driving the dual jack screws to provide elevation control. The entire elevation gear train is original equipment and has no nameplates or documentation to provide details of manufacture or specifications. Located on the left in the controller cabinet is the azimuth drive, a Copley Xenus XTL (8) driving a Baldor 3 KW motor connected to the azimuth transmission by a chain drive unit. The azimuth gearbox and the bull gear are original equipment and have no nameplates or documentation to provide details of manufacture or specifications.

Figure 13: Azimuth Motor controller and Elevation Variable Frequency Drive

3.0 Position Indication System Development (Software)

The Dish Control Software (DCS) was written to provide a software interface between the Mount Axis encoders (Azimuth and Elevation) and a GUI display to provide AZ/EL and Right Ascension (RA) and Declination (DEC) position data. Additionally, an Axis Hardware Simulator was developed to model the Haswell encoder hardware to allow DCS software development and test in Colorado Springs with occasional software releases tested at the Haswell site.

The DCS software was written in Java, with additional Open Source Java Libraries to facilitate the development of software interfaces to the pedestal hardware. These packages included the RXTX Serial Communication Package for Java, which provides a Serial Port Communication Library, the PI4J General Purpose I/O (GPIO) support library, which allows Java programs to communicate to external hardware, via the Raspberry PI GPIO bus. For the DCS development effort, this package allows the software to communicate with the 16x2 Line LCD Display. Additionally, other packages were used for the development of the GUI display software.

At this point in the development, there are a small number of mathematical algorithms used for coordinate and time conversions. Many of the implemented algorithms can be found in Jean Meeus' book - *Astronomical Algorithms* (9) and Peter Duffett-Smith's book - *Practical Astronomy with your Calculator* (10). The major algorithm in the current version of the DCS software is the *Azimuth/Elevation to Right Ascension/Declination* coordinate conversion routine.

3.1 Dish Control Software (DCS) – Top Level Capabilities

As stated above, the top-level capabilities provided by the DCS is a software interface to the Axis encoders and the display of the position data (See figure 14 for a detail block diagram of the Hardware/Software interfaces). A slightly larger list of capabilities includes:

1. Provides a software interface to the Antenna Axis Position Encoders.
    - Interface to each Encoder is via an individual serial communication port.
    - Each axis encoder can return a value between 0 – 4095.
    - The DCS interprets the encoder position data to generate Azimuth and Elevation positions.
    - The System.Properties file contains Encoder to Position Mapping constants.
2. Provides a Graphical User Interface (GUI) to display the AZ/EL and encoder position data.
    - Additionally, the GUI displays a calculated RA and DEC position based on the current AZ/EL position, location and time.
3. Provides a 16x2 Character LCD display of the current AZ/EL Position.
    - The LCD display also provides an operator interface to shut down the Raspberry Pi.

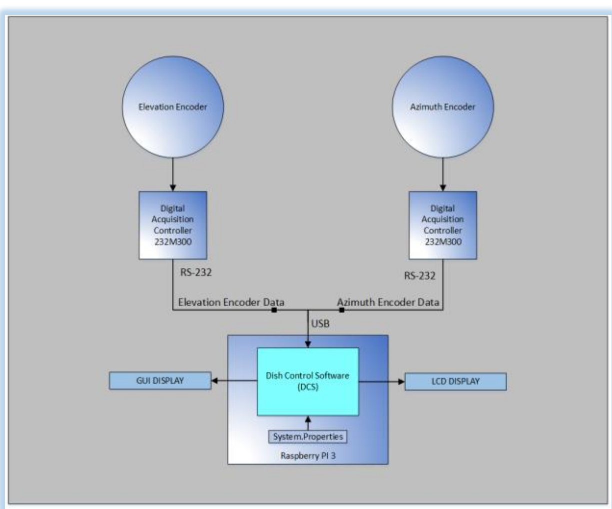
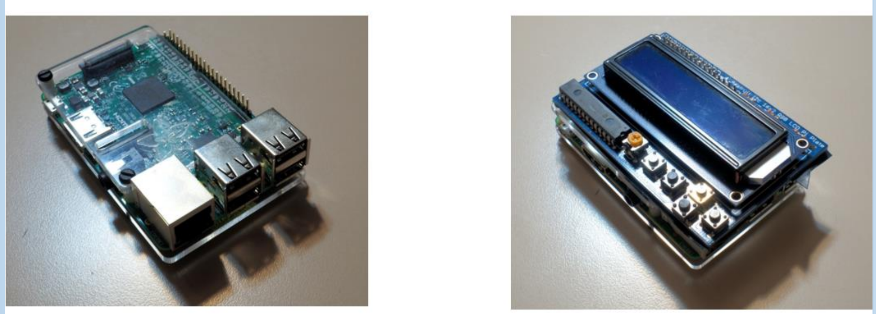


Figure 14: Hardware/Software Block Diagram

3.2 Development Environment

The development and target computer systems are both Linux based computers. Through the use of the Java development tool-kit, the team was able to develop and test on a low-end Dell Workstation, then move the compiled Java "jar files" to the target Raspberry PI for execution and test (See figure 15 for representative Raspberry PI hardware). The system development/test environment consists of:

1. A Dell Workstation running the Mint Linux 17 Operating System.
2. A Raspberry PI 3 Model B running the Raspbian Linux Operating System.
3. An USB to Serial Adapter cable (2 RS-232 Ports) to provide serial communication from the Encoders/DACs to the Raspberry PI.
4. On-Board 10/100 Mbit/s Ethernet Network Interface Card (NIC) is used for Ethernet communications (Used for X Forwarding of Displays).
5. Oracle's Java 1.7 Software Development Kit (SDK).
6. Eclipse Integrated Development Environment (IDE) - Used for Software Development and Unit Level Testing.
7. Major Java Support Libraries including:
    - PI4J General Purpose I/O (GPIO) support library – Provides Hardware Communication to the 16x2 Line LCD device.
    - RXTX Serial Communications Support Library - Enables access to Serial Interfaces in Java Applications.
    - Java Swing Library – Provides GUI routines required for GUI Window Development and Display.
    - Java LCD support library – Layers software support on top of Pi4J for displaying character data.



Raspberry Pi 3 Model B   Raspberry Pi 3 with LCD Display

Figure 15: Rasberry Pi 3

3.2.1 The Axis Simulator (ASIM)

The Axis Simulator was developed to provide a Development/Test Environment separate from the Haswell Deployment Site. The ASIM main function is to replicate the serial communication path between an individual Encoder/DAC and the DCS software. The message traffic consists of "String" data which is passed between a DAC and the Raspberry PI. On a periodic timer, the DCS software requests current position data from each of the encoders. This request "String" consists of a capital character "I" followed by a Carriage Return "\r". Each DAC, in response, returns a capital character "I" followed by "xxxf\r" where xxx is the hex value of the current encoder position (See figure 16 for a graphical depiction of the software message interface). The DCS software converts the Hex String to an integer encoder value, then converts the encoder data into AZ/EL and RA/Dec position data.

The Simulator provides the following capabilities:

1. Simulates each of the Axis (AZ/EE) and generates the appropriate serial data to simulate the RS-232 data processed by the DCS
2. Provides a GUI interface which displays current AZ/EL and Encoder Information
    - Allows the Developer to independently slew each Axis (Simulating the Hand Paddle)
    - Allows the Developer to verify the DCS AZ/EE Display Data
    - Allows the Developer to check edge cases (i.e. Slewing over Zenith, Pedestal Wrap …)

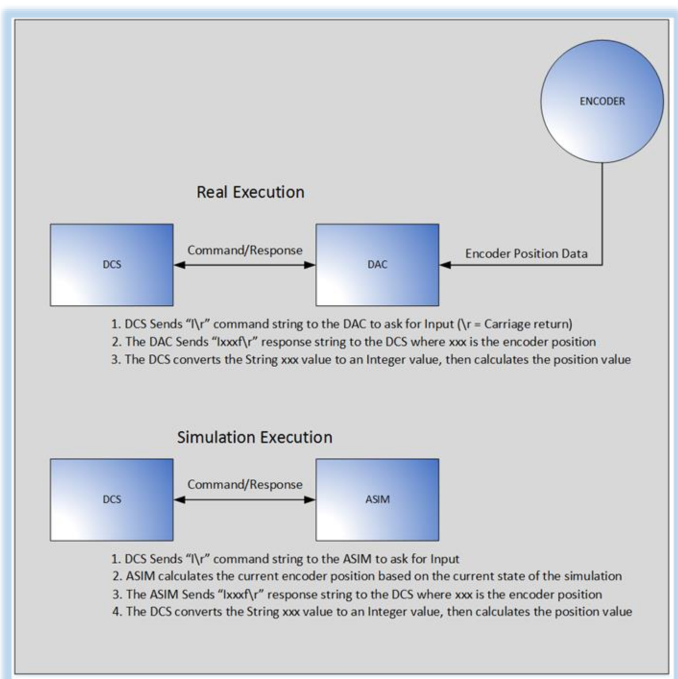


Figure 16: Real vs. Simulation Block Diagram

3.3 Developed Java Classes

The current DCS and ASIM software was developed using the Java 1.7 Software Development Tool-Kit (SDK). Below, the major classes in the two system are listed and described. Figure 17 provides a modified Class Diagram which shows a "logical" placement of the individual classes, and the interaction between the DCS and ASIM software components.

3.3.1 DCS Software Classes

- DishMountGUI – System "main". Builds and displays the DCS GUI then starts the main processing threads (DishMountInterface).
- DishMountInterface – Starts up the LCD Display and the Azimuth and Elevation processing threads then queries each axis thread every 250 ms to update the DCS GUI position data.
- MountAxisThread – A thread for each axis. This thread commands the DAC, via the serial port, for encoder position data, it then reads the position data from the serial port. The encoder position data is then converted to an AZ or EL Position. The encoder position and AZ/EL position are saved internally for display processing. The AZ/EL position data is converted to Ra/Dec for display.
- SerialPortDevice – A class for each serial port. This class manages the initialization and the software interface to a serial port (read/write). The system has two serial ports, one for each DAC/Encoder.
- LcdDisplayGUI – This manages the initialization and the software interface to display position data on the 2 Line LCD Display.

3.3.2 ASIM Software Classes

- AxisSimulatorGUI – System "main". Builds and displays the ASIM GUI then starts the main processing threads (AxisSimulator).
- AxisSimulator – This class manages the startup of the two axis simulation threads (AxisSimulationThread) and the startup of the two axis threads (AxisThread).
- AxisSimulationThread – This class contains a thread that models an Axis (encoder) position and movement. There is a AxisSimulationThread object for each axis. Axis movement is simulated based on movement commands initiated from the ASIM display.
- AxisThread – This class contains a thread that interfaces with an Axis serial port. It accepts "I" commands from the DCS and uses it's associated AxisSimulationThread object to respond with the current encoder position data. The response is in the form "Ixxxf\r" where xxx is the "hex" encoder position.

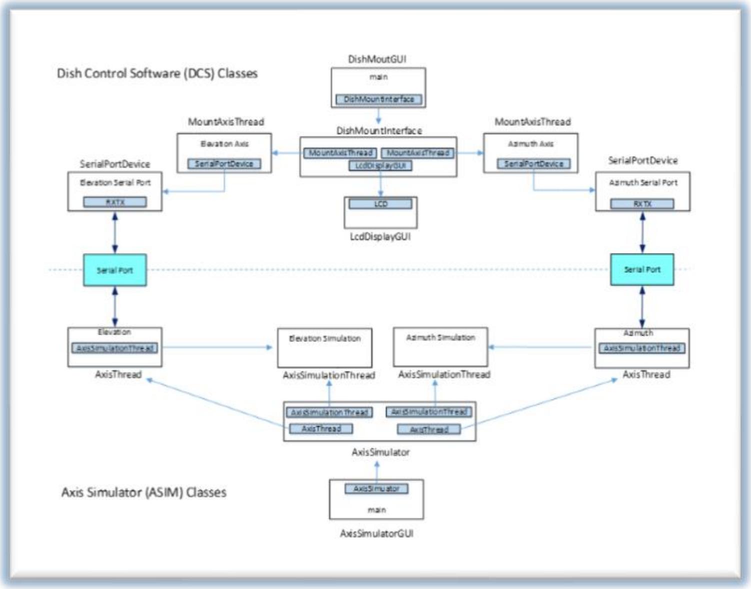


Figure 17: JAVA Software Classes

3.4 DCS and ASIM Displays

The DCS GUI Display provides AZ/EL and Ra/Dec position data at the current time. Additionally, the DCS provides a LCD Display which can be used in the Dish Pedestal to allow the operator to point the dish while being co-located with the pedestal hardware. Figures 18 and 19 depict representative displays associated with the DCS and ASIM software.

The ASIM display permits a software developer to move the "simulated" dish in both axis to allow testing of the DCS software. This capability mimics the Hand Paddle which controls each axis at the site. Additionally, Encoder and AZ/EL position data are displayed.

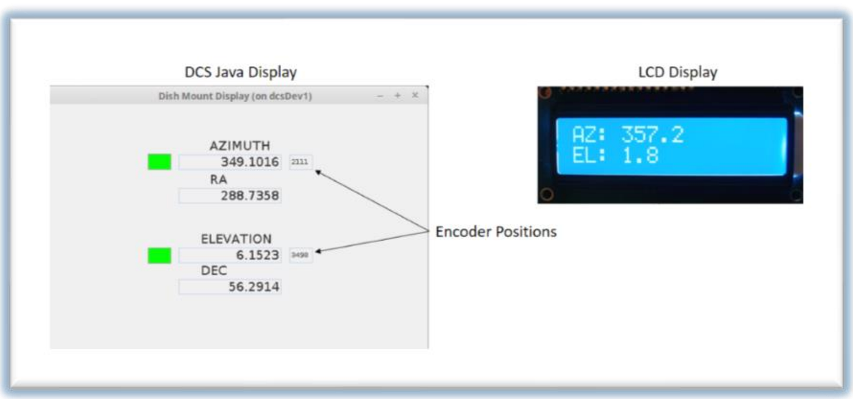


Figure 18: DCS JAVA and LCD Displays

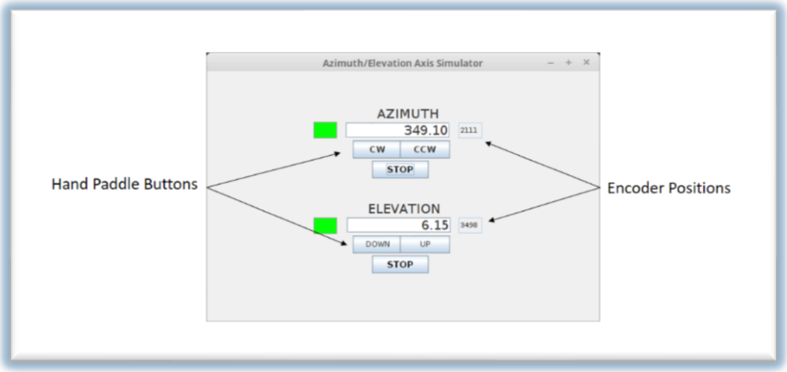


Figure 19: ASIM Display

3.5 System.Properties File

The system.properties file provides initialization data for the DCS software system. When the system is initialized, the system.properties file is read. Constants from the file are used to initialize the serial ports to allow serial communication, program the display update rates, and translate encoder positions to AZ/EL and RA/DEC position data. It contains:

1. Serial Port Configuration Parameters – Port Names and Baud Rates for each of the serial ports which communicate with DAC/Encoders.
2. Display Update Rates which determine how often the DCS GUI Display is updated.
3. Azimuth Translation Parameters – Start and Stop Positions, Encoder "Zero" Position for Azimuth.
4. Elevation Translation Parameters – Start and Stop Positions, Encoder "Zero" Position for Elevation.

3.6 Software Execution

The DCS software executes on a Raspberry PI single board computer which is located in the antenna pedestal, the GUI Display data is shown on a laptop located in the Communications Trailer (figure 20). The display data is passed, via the ssh X11 Forwarding protocol, to the laptop. X Forwarding is accomplished, via logging into the Raspberry PI, from the laptop using the Linux ssh command (i.e. ssh –X pi@192.168.178.22). Once the operator is logged in, the DCS software is started and the GUI display data is automatically redirected, via Ethernet, back to the laptop. Using the display data and the hand paddle, a system operator can point the dish to a selected AZ/EL or Ra/Dec position.

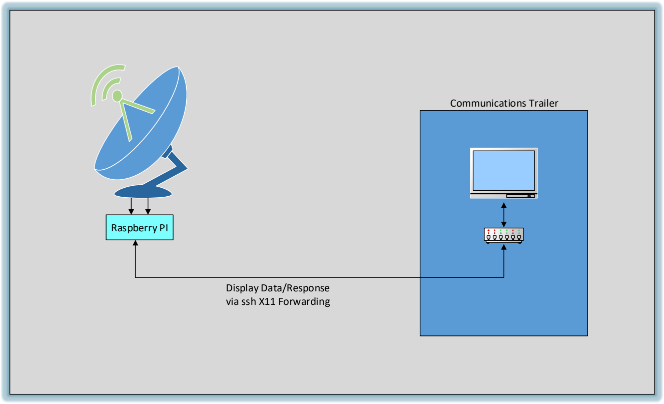


Figure 20: DCS Network

3.7 Software Future Development

The DCS software development will be a long-term development effort. An initial capability has been provided to allow for accurate pointing of the dish. Near term modification include:

1. Separate the GUI software from Axis Interface software.
    a. Allows the GUI software to run from any device with communication to the Raspberry PI via TCP/IP.
    b. Allows the use of more accurate time to be used in position calculations. The Raspberry PI does not have a Real-Time Clock (RTC) so the calculation of AZ/EL and RA/DEC will be completed on a client system that has more accurate time.
2. Add Additional Display Capabilities.
    a. Graphical Display showing the current Axis Position in RA/DEC space.
    b. Pulldowns with lists of known radio source objects.
    c. Operator control of the Raspberry PI in the Pedestal. Remote reboot and shutdown capabilities.
3. Create standalone tools to support observing, data collection and system testing.

Long term modifications will include commanding and real-time tracking of the antenna via software hardware control algorithms.

4.0 System Testing

Testing of the Mount hardware and the DCS software was completed in multiple phases. The first phase, Developmental Testing, was completed in a workbench environment with a single Encoder and the Signal Conditioner Board /Data Acquisition Boards. In this environment, the ability to communicate with an Encoder was established and tested.

The second phase, Site Integration Testing, was completed at Haswell. During this phase the hardware and the software were integrated and installed at the Haswell site. Using known visible landmarks, azimuth positions were calculated for each of the landmarks. The dish was then pointed at the landmarks and sighted and the encoder positions were captured. Using the calculated azimuth positions and the encoder positions, a "zero" azimuth/encoder relationship was established. The elevation "Zero" encoder relationship was also established by pointing the Dish at the zenith, 45 and 90 degree positions. The system.properties file was then updated with the Position/Encoder initialization values.

The final phase, Beacon Testing, was also conducted at the Haswell site. In this testing, known radio beacons and their azimuth position (with respect to the Haswell site), were used to verify the azimuth positioning. Additionally, this testing utilized the recently completed acquisition electronics (Feedhorn) for a "first light" type test.

4.1 Developmental Testing

The initial developmental testing was completed using the mount hardware in a stand-alone environment. A single encoder was connected to the Signal Conditioner Board which in turn was connected to the Digital Acquisition Board. The Digital Acquisition Board contains the DAC which provided encoder data, via a serial interface, to the DCS. Located on the Signal Conditioner Board are 12 LEDs that display the current data from the encoder. These LEDs provided the "expected" results during development/test of the Encoder Interface Java classes (the MountAxisThread class). To facilitate testing, the Encoder can be "hand" turned which changes the reported encoder position data. To support testing, the Java software was "instrumented" with debug statements, which allowed the developer to inspect the LEDs and the reported data from the DCS software. Using this method, the developer was able to verify the range of position data, versus position data reported by the software.

4.2 Site Integration Testing

Once developmental testing was completed, the Mount hardware and DCS software were deployed at the Haswell site. For azimuth testing, three survey point were determine using Google Maps. These points were fixed azimuth positions (landmarks) located on the local horizon. The dish was slewed to these fix points and the position data was recorded. Positioning was accomplished, via line of sight pointing, using an antenna support arm as the pointing vector. Repeated slews were used to obtain the data and to allow for verification of the "repeatability" of the encoder readout. Based on the consistency of the reported positions, it was verified that the azimuth encoder readout was working correctly. Based on the calculated azimuth positions of the "fixed" targets, a "zero" azimuth position was calculated and stored in the system.properties file.

The elevation axis Position/Encoder relationship was established using a level and slewing the dish to the zenith (0), 45 and 90 degree positions. When the axis was level at these three positions, using the mount support structure as the level mechanism, the encoder positions were read-out and the "zero" elevation position was calculated and stored in the system_properties file. Table 1 shows some of the results from the integration testing. The calculated data is based on an encoder reporting 0 – 4095 counts and a single encoder revolution equal to 360 degrees. A single encoder "tick" is equal to 360/4096 or 0.087890625 degrees/count. Counts/degree would be equal to 4096/360 degrees or 11.377777 counts/deg. So as an example, the elevation value of 45 degrees can be calculated by: 3587 – (45 – 0) * 11.377777 □ 3075 counts.

Elevation	0	45	90						
Calculated	3587	3075	2563						
Observed	3587	3075	2568						
Azimuth	0	45	54.21	90	135	180	225	270	315
Calculated	1975	1463	1358.2107	951	439	4023	3511	2999	2487
Observed	1975	?	1366	?	?	?	?	?	2453

Table 1: Test Data

4.3 Beacon Testing

When the Dish's Feedhorn hardware became available, known radio beacons were identified and their site azimuth positions were calculated. The Dish was slewed to each of the positions and a radio receiver was used to establish and verify their reception (at their expected frequencies). The displayed azimuth position was compared to their calculated position. In each case (3 beacons were utilized) the reported azimuth position and the calculated positions were within 1 degree of each other.

In the coming months, additional testing to support both AZ/EL and RA/Dec verification will use the Earth-Moon-Earth communication (EME) technique, also known as moon bounce, to verify the hardware/software capability to point the Haswell Dish at a fixed right RA/Dec position.

5.0 Future Control Capability

The planned future capability is to automatically point and track an astronomical object. Figure 21 shows the preliminary control board. The operator will be able to enter a right Ascension and Declination, and then the control system will maintain pointing at that position through emulation of the manual control paddle through a series of computer controlled relays. The system will also enforce safety issues such as maximum/minimum azimuth or elevation angle stops and cable wrap turns.

Another future concept is to conduct active tracking on signal. This will allow the antenna to track on the signal path which may be different than the programmed signal path.

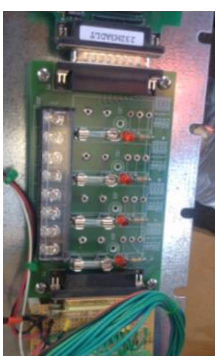


Figure 21: Antenna Motor Controller Prototype Board

6.0 Summary

The 60-foot antenna position indicator system is vital to provide scientific grade observations for radio astronomy. The system is based on the use of 12-bit encoder accuracy that is converted to azimuth and elevation as well as right ascension and declination. The outputs are displayed at the operator station in the communications trailer. The position data is combined with the output data to provide scientific grade radio astronomy observations.

DSES will use the pointing system for astronomical observations as well as EME and tropospheric radio communications.

References

1. *Plishner Radio Astronomy and Space Science Center 60-Foot Dish Multi-Band Feed Development.* Uberecken, Ray, Babcock, Myron and Russel, Richard. St. Louis, Missouri : ARRL, October 13-15, 2016, Proceedings of Microwave Update 2016. ISBN: 978-0-87259-665-8.

2. https://www.amci.com/files/3514/5463/6666/rotary-encoder-absolute-digital.pdf . [Online]

3. http://www.integrityusa.com/232m3adlt/ . [Online]

4. https://www.adafruit.com/products/1115. [Online]

5. https://www.adafruit.com/products/3055 . [Online]

6. http://xppower.com/pdfs/SF_ECP40.pdf. [Online]

7. http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1305-um002_-en-p.pdf . [Online]

8. http://www.copleycontrols.com/Motion/pdf/XenusUserGuide.pdf . [Online]

9. Meeus, J. *Astronomical Formulae for Calculators.* s.l. : Willmann-Bell, Inc., 1988.

10. Duffett-Smith, P. *Practical Astronomy with Your Calculator.* s.l. : Cambridge, 1995.

11. Meeus, J. *Astronomical Algorithms.* s.l. : Willmann-Bell, Inc., 1998.

12. S. Urban, P. Seidelmann. *Explanatory Supplement to the Astronomical Almanac.* s.l. : University Science Books, 2013.