SPHERES/Astrobee Working Group (SAWG)
Quarterly Meeting
Aug 23rd, 2017
Meeting Kick off
Meeting Logistics

- **Topic:** SPHERES/Astrobee Working Group (SAWG) Quarterly Meeting
- **Purpose:** Information sharing across the SPHERES & Astrobee community - not intended to be project reviews!
- **Date:** Wed, Aug 23rd, 2017
- **Location:** Virtual
- **Time:** 8:00 am, PST

JOIN WEBEX MEETING
- [https://nasa.webex.com/nasa/](https://nasa.webex.com/nasa/)
- Meeting number: 392 669 018
- Meeting password: Astrobee!8
- Join by phone: 1.844.467.6272
- pass code: 445068
<table>
<thead>
<tr>
<th>Time</th>
<th>Event Description</th>
<th>Duration</th>
<th>Presenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:00am</td>
<td>SPHERES Welcome</td>
<td>2 min</td>
<td>Andres Martinez</td>
</tr>
<tr>
<td>8:05am</td>
<td>SPHERES Facility</td>
<td>10 min</td>
<td>Jose Benavides</td>
</tr>
<tr>
<td>8:15am</td>
<td>SPHERES Ops</td>
<td>15 min</td>
<td>Aric Katterhagen</td>
</tr>
<tr>
<td>8:30am</td>
<td>SPHERES Eng</td>
<td>15 min</td>
<td>Jonathan Barlow</td>
</tr>
<tr>
<td>8:45am</td>
<td>SPHERES PIM</td>
<td>10 min</td>
<td>Melissa Boyer</td>
</tr>
<tr>
<td>8:55am</td>
<td>Astrobee</td>
<td>15 min</td>
<td>Maria Bualat</td>
</tr>
<tr>
<td>9:05am</td>
<td>Astrobee Sim</td>
<td>5 min</td>
<td>Andres Mora Vargas</td>
</tr>
<tr>
<td>9:10am</td>
<td>VertigoSmoothing</td>
<td>10 min</td>
<td>Danilo/Alvar</td>
</tr>
<tr>
<td>9:20am</td>
<td>Zero Robotics</td>
<td>10 min</td>
<td>Katie Magrane</td>
</tr>
<tr>
<td>9:30am</td>
<td>SLOSH Lessons</td>
<td>10 min</td>
<td>Brandon Marsell</td>
</tr>
<tr>
<td>9:40am</td>
<td>Tether Slosh</td>
<td>10 min</td>
<td>Hans Zachrau</td>
</tr>
<tr>
<td>9:50am</td>
<td>SVGS–RINGS</td>
<td>10 min</td>
<td>Hector Gutierrez</td>
</tr>
<tr>
<td>10:00am</td>
<td>Metis–RFID</td>
<td>10 min</td>
<td>Jose Cortez</td>
</tr>
<tr>
<td>10:10am</td>
<td>HoneyBee–UDI</td>
<td>10 min</td>
<td>Jack Wilson</td>
</tr>
<tr>
<td>10:20am</td>
<td>Tethers–CobraBee</td>
<td>10 min</td>
<td>Nathan Britton</td>
</tr>
<tr>
<td>10:30am</td>
<td>Altius</td>
<td>10 min</td>
<td>Jonathan Goff</td>
</tr>
<tr>
<td>10:40am</td>
<td>Stanford–Gecko</td>
<td>5 min</td>
<td>Abhishek Cauligi</td>
</tr>
<tr>
<td>10:45am</td>
<td>Illinois–Elecro</td>
<td>5 min</td>
<td>Matthew Spenko</td>
</tr>
<tr>
<td>10:50am</td>
<td>Maryland–Gripper</td>
<td>5 min</td>
<td>Christine Hartzell</td>
</tr>
<tr>
<td>10:55am</td>
<td>Concluding remarks</td>
<td>5 min</td>
<td>Jose B. Realm</td>
</tr>
</tbody>
</table>
• A Facility of the ISS National Laboratory with three IVA nano-satellites designed and delivered by MIT to research estimation, control, and autonomy algorithms
• Installed on ISS in 2006
• Managed by ARC since Fall 2010
• By working aboard ISS under crew supervision, it provides a risk tolerant Testbed Environment for Distributed Satellite & Free-flying Control Algorithms
  ✓ Formation flight, Docking, Proximity Operations
• If anything goes wrong, reset and try again!
• The satellites can be reused
  ✓ Replenishable consumables
  ✓ Multiple test sessions assigned per year

If you can’t bring the space environment to the laboratory, take the laboratory to space!

Over 121 Test Sessions (600+ hrs. of Facility Console activities involving crew)
One of the most used and popular ISS National Lab Facilities
SPHERES Community

- **SPHERES Working Group (SWG) Quarterly meeting**
  - Membership includes MIT, FIT, AFS, DARPA, CASIS, SJSU, Airbus, and NASA (HQ, KSC, JSC, MSFC, and ARC)
  - Face-to-Face, twice a year
  - Next Face-to-Face will be scheduled in Nov. 2017 at NASA Ames

- **Purpose:**
  - Information sharing across the SPHERES/Astrobee community
  - Program office shares
    - National Lab Facility availability
    - Status of resources (batteries, CO2 tanks, etc.),
    - Overall Calendar (scheduled Test Sessions, upmass/return), and
    - Updates on “new” PD, Investigations, and ISS infrastructure.
  - Provide the SPHERES/Astrobee community (PD, investigators, etc.) with up-to-date information to determine opportunities to use the NL Facility
  - Discuss proposed changes/updates to SPHERES Nat Lab which may be required to support a specific activity or research.
  - Discuss specific support requests made to the ISS Office
SPHERES Facility Team

Team

- Jose Benavides, Jose.V.Benavides@nasa.gov, PM
- Aric Katterhagen, aric.j.katterhagen@nasa.gov, Ops Lead
- Jonathan Barlow, jonathan.s.barlow@nasa.gov, Eng Lead
- Jose Cortez, jose.cortez@nasa.gov
- Robert Hanson, robert.s.hanson@nasa.gov
- Simeon Kanis, simeon.i.kanis@nasa.gov
- Don Soloway, Donald.i.soloway@nasa.gov
- Andres Mora Vargas, amora@meicompany.com
One Man Down
Qty 65 gen 2.5 CO2 ready for shipping by 8/31
Building a light shade for new ISS lighting that allows SPHERES to continue operating in presence of IR noise
ISS Beacon repair done, launching on OA-8
Last Inspire-2 MIT Halo Test Session June 23rd
Inspire-2 investigation completed with workshop on June 21st
Successful Zero Robotics Field Day at Ames, July 7th
Successful Zero Robotics Finals competition on Aug. 11th

NASA astronaut Jack Fischer helps conduct an in-space competition called SPHERES Zero Robotics that challenges middle and high school students to guide a bowling-ball-sized satellite around the interior of the space station.
Tether-SLOSH kickoff May 23rd
Vertigo/"Smoothing-Based Relative Navigation” Kickoff
Astrobee project support continues
Astrobee Robotics Software Simulator Beta release & open-sourcing targeted 8/31
Guest Scientist Guide 8/31
Mechanical Payload ICD 8/31
Two presentations, ISS R&D conference July 17th
Integration & Test
Flight Build
Supported winning poster at Innovation Fair: “Ames K-12 STEM Challenges”
Summer 2017 Interns: Qty 8

- Aris Koumis & Dean Yuan: Astrobee Microbial Sampling Payload (AMSP)

- Eddy Meza: SSLA Shade
Elizabeth Nguyen: Astrobee Payload Expansion Port Tester (APEPT)

Jacob Killelea: Multi-robot Communication

BASIC DESIGN REQUIREMENTS

1. The system shall establish USB connection with the Astrobee using the USB data pins on the payload interface.
   a. The system should identify the Astrobee as a USB device.
   b. The Astrobee should identify the payload as a device.
   c. The system should be able to transmit and receive data.
2. The system shall test and demonstrate current and voltage outputs.
   a. The system should show a possible current draw of 3A.
   b. The system should show voltage levels of ~14.4 volts.
   c. The system should limit the current draw of the payload.
3. The system shall be able to attach to the payload interface using both the lever method and the screw method.
4. The system should have on board computing capabilities for guest science/test software.
5. The system should enhance the mechanical mounting of the payload interface to allow for additional attachments. (Bonus)

Payload Connector Pin Types

- Power & Ground
- MLP USB D+/- x 2
- HLP USB D+/-
- Reserve Pins [I2C/UART]

APEPT HARDWARE DEV

I. Electrical
   The development of the hardware for the APEPT was centered around the use of the BeagleBone Black (BBB) - a single-board computer/microcontroller; chosen for its easy-to-use development environment, low cost, and flexibility. The Astrobee PCB layout was for the BBB to organize and provide the additions to run the port interface.
   The APEPT was linked to a CAD model for the Astrobee. The model was a potential unit, LCD display, and

APEPT SOFTWARE DEV

For the software development of the APEPT, several different frameworks were studied:

1. Robot Operating System (ROS)
2. Android OS APK development
3. JavaScript + BoneScript + NodeJS

The ROS and Android framework were chosen because Astrobee currently uses these frameworks for operation. The JavaScript + NodeJS was chosen by recommendation and BBB community tutorials.

Each of these frameworks are implemented on either the APEPT or Astrobee to conduct the port tests and serve as example use cases/implementation guides for guest science.
Summer 2017 Interns: Qty 8

- Matt Moropoulos: SPHERES-Hook
- Michael Richardson: MGTF Control Software
- Rachel Crum: Payload to Ground Communication

**Approach & Progress**

1. Understanding DDS
   - Developed messaging protocols, specifically DDS.

2. Implementation Design
   - Decided upon Python for its great documentation, readability, ease of use, and consistency with the visualizer.
   - Established that a reader/sensor would subscribe to DDS messages and send the message to the visualizer.

3. Practice Makes Perfect
   - Developed reader and writer scripts that send messages to each other.
   - Developed a reader script that subscribed to the Actinometry simulator and received position data.

4. Separate ROS from the Current QMC Visualizer Code
   - Two separate ROS and DDS scripts that would be triggered based on a user-defined flag.
What’s next …

- Next ZR competition is under way
- New Vertigo Smooth Navigation research
- Tether-Slosh
- Continue work transitioning to Astrobee
  - Goal: Fully operational in 2018
Guest Science Program (GSP)

- What's available from the Astrobeel Facility?
  - Astrobeel Robotics Software Simulation
  - Ground Hardware: Qty 3 & ”Flat-Sats”
  - Labs: Granite & MGTF
  - Documentation and Training
  - Proposal Support
  - ISS Payload Partner

- How can I use Astrobeel and what does it take?
  - Guest Scientist Guide & Mechanical Payload ICD
  - New Hardware or ”just” Software?
  - Ground Demonstration or ISS Operation?

- We want to hear from you!
  - Approximate Scheduling
SPHERES & Astrobee Operations
Operations: Functions

Ensure Facility Readiness for ISS Test Sessions

- All crew training now via Onboard Training (OBT) both English & Russian
- Crew procedure updates
- Coordinate with ISS Lead Increment Scientist and POIC Cadre
- Flight products on orbit (test plan, .spf, on-board training and review, etc.)
- Consumable (CO2 Tanks and Batteries) refurbishment and resupply
- Support SPHERES directory/file maintenance

Real-Time ISS Test Session support

- Coordinate w/SPHERES investigators product development and delivery
- Support crew and POIC cadre real-time
- Conduct/coordinate crew conferences as needed
- Test session data and video management

Public Relations

- Maintain website, work with ARC PAO office to publish material on site
Operations: Functions

Increment Planning

- PTP and 2-pager development, and update & support and submittal
- Timeline planning model review and update

Safety and Verification Assessments

- Integrated Safety & Verification Assessments for all SPHERES payloads
- Safety & Verification assessments for Battery/Tank launches/returns
- Complete Certification of Flight Readiness for ground systems and on-orbit hardware and operations products
- Conduct ISS Requirements Change Assessments to SPHERES Facility

Astrobee Ground Ops Development

- On orbit Activity planning and development
- Ground Operations Readiness Test planning and development
- First ISS Ops - Mapping Activity being planned, procedure in final stages
Ops: Increments 51/52 Review

Increments 51/52 (April 2016 to Sept 2017)

- Increment planning on going for 53/54 (Sept 2017 – March 2017)
  - Planning for Tether-Slosh, Smoothing-Based Relative Navigation, Zero Robotics
- Planning also for potential Slosh Coating, and follow on Docking Port ongoing
- Consumables and hardware manifest support for SpX-11, OA-7, 67P and SpX-12 safety, verification for these flights – summarized on later Ops slide and in PIM presentation
- Supported InSPIRE II Workshop at NASA HQ June 21
- Procedure and hardware development supported at MIT July 6-7
- Supported and presented ISS Conference in Washington D.C.
Ops: Increments 51/52 & 53/54


- Halo Science 1
  - June 23 2017
- Zero Robotics Unit Test/Dry Run
  - Aug 4, 2017
- Zero Finals
  - Aug 11, 2017

Tether-Slosh
- TBD

Smoothing-Based Relative Navigation
- TBD

Zero Robotics High School Units Test
- Nov- Dec 2017

Zero Robotics High School Dry Run
- Jan 2018

Zero Robotics High School Finals
- Jan 2018

Additional Docking Port
- TBD

Slosh Coating
- TBD
SPHERES Calendar
Consumables Status

CO2 Tank Inventory
- X Tanks on orbit now
- Planning to up mass pink tanks fall/winter 2017-18 – flight TBD
- Total: X gray tanks

Battery Pack Inventory
- X Batteries on orbit now
- 10 SPHERES Rechargeable Batteries arrived on station with OA-7

Consumables downmass
- 13 empty gray tanks to be returned on SpX-12
SPHERES on Social Media

Twitter
https://twitter.com/NASA_SPHERES

Website
http://www.nasa.gov/spheres
Astrobee System Overview

SPHERES/Astrobee Working Group
August 23, 2017
System Data Flow Diagram

Link Legend

- **Ethernet/LAN**
- **Ethernet**: Internal IP
- **Ethernet**: Internal and Payload LAN
- **WiFi**: Payload LAN
- **USB**
- **Other/LAN**

GS Inst  ARC MMOC  MSFC  JSC MCC

8/23/2017  SPHERES/Astrobot Working Group
Astrobee

- Speaker/Microphone
- Laser Pointer
- HazCam
- Touch Screen
- Power Switch
- Forward Flashlight
- Wake Button
- Status LEDs
- SciCam
- NavCam
- Nozzle
- Terminate Button
- SpeedCam
- Impeller
- Battery
- Arm
- DockCam
- PerchCam
- Aft Flashlight
- Corner Bumpers
- Signal Lights
- ScimCam
- NavCam
- Nozzle
- Terminate Button
- SpeedCam
- Impeller
- Battery
- Arm
- DockCam
- PerchCam
- Aft Flashlight
- Corner Bumpers
- Signal Lights
**Payload Attachment Options**

**Quick “No Tool” Payload Attachment**

- **2X Lever (open position)**
  - Lever engages and disengages payload connector and provides mechanical attachment.

- **Lever in “Locked” position**

**4X Fastener Payload Attachment**

- **4X Captive Fasteners**

- **“Un-Lock” Position**

- **“Lock” Position**

8/23/2017  
SPHERES/Astrobee Working Group
Dock

Air Vent Deflector
Subsystem Breakers
Main Power Breaker
Cooling Fan
Cooling Fan Screen
RJ-45 Connector
Power Connector
Free Flyer Berth
AR Target
Plan Editor Tab (Operator/Engineering)
Run Plan Tab (All)

- Select valid plan and upload to Astrobee
- List view of loaded plan
- Model of loaded plan
Teleoperation Tab (All)

Buttons here can be changed via config file

Drag preview to adjust movement command

Construct movement command

Adjust settings

Send movement command
Guest Science Tab (Crew)

- Checkboxes to select Astrobrees to command
- Status summaries
- Names of loaded Plans
- Monitor Astrobbee positions in 3D window
- Command Astrobrees
Advanced Guest Science Tab
(Operator/Engineering)

Start and Stop APKs directly
Preview and change APK command before sending
Select APK to see Status
View detailed telemetry from APKs
Advanced Tab (Engineering)

- **Detailed Health and Status**
- **Triggered and Not Triggered Faults**
- **View and change Operating Limits**
- **Detailed battery status**
- **Configure telemetry sent to Control Station**
- **Disk usage**
- **View and configure data saved to disk**

Detailed component status
Astrobee Status

• Finalizing drawings
• Procurement has begun
• On track for August 31st delivery:
  • Beta release of Flight Software/Simulator
  • Mechanical Payload ICD drawings
  • Initial draft of the Guest Science Guide
Guest Science and Simulator Integration
Andres Mora
Simulator

• Robot:
  – Hardware includes multiple cameras, processors, actuators, and mobility mechanisms.
  – Software allows localization within ISS, communication, grasping, actuation, locomotion.
  – Runs Guest Science payload (both hardware and software)

• Simulator:
  – Has the same code base as that in the robot
  – Adds components to simulate the environment inside the ISS and the communication with ground control
  – Allows users (Guest Scientists) to quickly test their experiments and understand how the robot would behave.
Simulator

• Developers require:
  – Simulator able to run in a medium to high-end laptop (e.g. Nvidia Quadro K1100M – Dedicated Video Memory 2048MB GDDR5)
  – Linux environment (Currently Ubuntu distribution, LTS 16.04)
  – Integrate ROS, Android (Android Studio) operating systems
  – Simulator is based on C++, Simulink deployed C blocks
  – Developers in Android extensively use multiple programming languages: Java, XML
  – Provides 3D representation of the robot via Rviz and/or Gazebo
  – Provides dynamical, sensor models via Gazebo
Software Architecture

- **Three ARM processors to isolate guest scientist code, vision based localization and GNC loop, connected by 100Mbps network switch**

- **High Level Processor (HLP)** – Android, Quad core
  - Interface with Science Camera and Display
  - Encodes video with dedicated hardware
  - Runs guest science code

- **Mid Level Processor (MLP)** – Linux, Quad core
  - Runs absolute localization algorithms, obstacle detection, communications
  - Heavy processing power used by vision

- **Low Level Processor (LLP)** – Linux, Dual core
  - Runs 62.5 Hz: EKF and propulsion control loop
Guest Science interface scenarios

1. Basic API (Zero Robotics): handles high-level commands and uses Android environment only


3. Advanced API (University researchers): By-passes completely provided Java-ROS framework.
Guest Science Implementation

**Pure Java**

- Guest Science Application Code
- ARS Java API
- ARS Lib
- GS Mgr. Stub
- ROS Java

**Android (Emulator or HLP)**

- Guest Science Application Code
- ARS Java API
- ARS Android LIB
- ARS Android SVC
- ROS Java

- Guest Science Manager

Higher fidelity simulation with the Android platform emulated

Model can be used on Android too if packaging ROS Java + ARS Lib in App

Not available in first release

Pure Java Model can be used on Android too if packaging ROS Java + ARS Lib in App

Higher fidelity simulation with the Android platform emulated

Not available in first release
• Communication between different processes/threads
• Intercommunication Process (IPC)
• Services:
  – Scheduled
  – Started
  – Bound
• Messenger class to avoid complex AIDL implementations
Astrobee Robot Software (ARS)
SPHERES/Astrobee Working Group, 08/23/2017

Lorenzo Flückiger
Maria Bualat
Intelligent Robotics Group
NASA Ames Research Center
Lorenzo.Fluckiger@nasa.gov
ARS (Astrobee Robot Software) Overview

- ARS is deployed on 4 cell phone type processors (Astrobee + Dock) running Linux and Android
- Astrobee contains 7 distinct microprocessors with custom firmware + several microprocessors with COTS firmware
- Software deliverables includes:
  - Custom firmware(s)
  - Custom tailored Linux kernels
  - Linux and Android Operating Systems
  - ARS dependencies (third-party libraries)
  - ARS Source Code ⬅ Open Source
ARS Features

- Manage Astrobee sensing and actuation
- Localize and Navigate within the ISS
- Perform autonomous docking (+ return to dock)
- Perform autonomous perching
- Support teleoperation from ground
- Support plan based automated tasks
- Support “Guest Science” operations
- Support multi Astrobees communication
- Support hardware for multisensory human interaction
ARS Components

- **OS (Communication Framework)**
- **Localization**
  - Marker less Flying
  - Docking
  - Perching
- **Offline mapping for localization**
- **Pose Estimation + Propulsion Control (GNC)**
- **Executive**
  - Mode Management
  - Sequencer (Plan Execution)
- **Mobility**
  - Generates and validates trajectories
  - Performs collision detection
- **Fault Management**
- **Guest Science**
- **User Interfaces Support**
- **Simulator**
- **Platform Management and development tools**

---

**SLOC Directory SLOC-by-Language**

<table>
<thead>
<tr>
<th>Category</th>
<th>Language</th>
<th>C++</th>
<th>Ansic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1367280</td>
<td>20471</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td>1367376</td>
<td>20471</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14015</td>
<td>14956</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11923</td>
<td>7963</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9366</td>
<td>984</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14262</td>
<td>12351</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9803</td>
<td>14262</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6393</td>
<td>6279</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6760</td>
<td>4778</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3527</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1844</td>
<td>2867</td>
</tr>
</tbody>
</table>

**Submodules**

<table>
<thead>
<tr>
<th>Category</th>
<th>Language</th>
<th>C++</th>
<th>Ansic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1367280</td>
<td>20471</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14015</td>
<td>14956</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11923</td>
<td>7963</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9366</td>
<td>984</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14262</td>
<td>12351</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9803</td>
<td>14262</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6393</td>
<td>6279</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6760</td>
<td>4778</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3527</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1844</td>
<td>2867</td>
</tr>
</tbody>
</table>

**Totals grouped by language:**

- **C++:** 1422365 (96.05%)
- **Ansic:** 21455 (1.45%)
- **XML:** 14956 (1.01%)
- **Python:** 9792 (0.66%)
- **Java:** 7963 (0.54%)
- **Sh:** 2867 (0.19%)
ARS and ROS

- ARS makes extensive use of the open-source Robot Operating System (ROS):
  - Communication framework linking all “nodes” running on the target platform
  - Try to maximize the re-use of existing ROS messages benefit from existing ROS packages
  - Use ROS introspections tools to rapid debugging
  - Use ROS facilities to record/replay/analyze data
  - Use some ROS/Gazebo components for the simulator
Simulator

- Custom propulsion system and some localization sensors
- Gazebo based dynamics model, imagers, arm, lights and ISS model
- Can run all nodes on desktop or some nodes on target development board
ARS APIs Overview

• ARS uses ROS within Astrobee: Messages, Services and Actions define the internal API

• Astrobee & Ground communication uses DDS and the RAPID framework for command and telemetry

• Commands:
  – Commands are defined using XP-JS0N schema, tools auto-generates RAPID command dictionary
  – ARS defined a ”ROS Command” mirroring the DDS command structure
  – Onboard Astrobee Guest Science or Ground Applications share the same command dictionary (some commands unique to one client) with either DDS or ROS transport

• Telemetry:
  – Internal uses ROS Messages (using ROS messages when possible)
  – External uses DDS Messages (subset only, re-using RAPID messages)
Hardware Status

- Red: 100%
- Blue: 100%
- Black: 25%
- Orange: @ MIT
- Stack: 100%
- Battery recharger: 4 of 4
- New air carriage: 3 of 4
  - Tall 3-puck: 2 of 3
  - Single puck: 1 of 1
  - More in work
Ground Lab Status

- Granite Lab: Online
- Flight Lab: Online
- Micro Gravity Test Facility (MGTF) Lab
- Engineering Evaluation Lab (EEL): Available upon request
SPHERES ISS Beacon Repair

Inventory

- Corrosion discovered from Batteries
- Discovered by Jeff Williams during TS82 (Maintenance Session)

Results

- Arrived at Ames
- Battery holder replaced, cleaned, and inspected
- Returned to station
Gen 2.5 Pink CO2 Tank Status

- **Inventory Status**
  - ~60 Tanks Ready for Filling
  - Decals Arrived (IMS Barcode / Hazzard waste / Fill Status)

- **Schedule**
  - ~1 week to finalize product (fill, label, bake out) + 1 week Idle leak check
  - Full delivery mid September

- **Notes**
  - Under-Torqued pin-valve issues on Gen 2.0 has been resolved with new oversight and integration at Ames.
Lab Upgrades: MGTF

- **Environmental effects**
  - Background
  - Mock ISS Panels
  - LED Lighting
  - Active Gimbal
MGTF Gimbal

- Gimbal Specs
  - Active Gimbal
  - Yaw 360°  Roll 126°  Pitch 180°
  - Weight ~ 30lbs (without satellite)
SSLA Shades (Remember this?)

- JEM GLA lights are planned to be replaced soon by SSLAs
- SSLAs cause IR noise andre sets SPHERES
- Prototype testing at JSC early next month
Software: Gantry Control

6 dof Control:
- Matlab control software.
- Position, Velocity, or Acceleration commanding.
- Visualeyez for ground truth
- Real time commands from simulation or from the payload over Xbee.

Status:
- 6-DOF operational.
- Testing in progress