Outline

• Astrobot Overview
• Astrobot Interface
• Astrobot Development Tools
Astrobee Overview

- A total of 3 units on board ISS
- Docking station for recharge and wired comm.
- Built in perching arm using payload interface
- 6 total cameras for various purposes, including one cellphone class HD camera.
- Main purposes:
  - Host guest science payload (GSP payloads)
  - Serve as mobile camera for ISS situational awareness
  - Serve as mobile sensor platform for
- First GSP Payloads
  - REALM RFID reader
  - Zero Robotics High School and Middle School competitions
Three main internal processors

High-Level Processor (HLP)
- Snapdragon 805 Quad-Core SOM
- Mostly reserved for GSP payload use
- Runs touch screen and...

Mid-Level Processor (MLP)
- Snapdragon 805 Quad-Core SOM
- Lower rate, higher computation processes
- Runs most of the core robot software
- Handles decision making and data communication

Low-Level Processor (LLP)
- Wandboard dual
- High rate control loop and sensor sampling
- Pseudo-real time
Interface - Attachment

• Available volume:
  • 12.32 cm by 15.24 cm by 10.16 cm
  • Keep outs provided via CAD model
  • Payloads may extend beyond the exterior of robot
  • Payloads within payload volume use built-in bumpers for impact protection

• Alignment pins

• No-tool quick-release levers attachment system

• Four #8-32 bolt pattern for each GSP payload bay
Interface - Electrical

• Connector:
  • Astrobee side: Glenair M83513/03-E03N
  • Payload side: Glenair M83513/04-E03N

• Power:
  • 14.4 V 3A unregulated

• Data:
  • USB to HLP and MLP
Interface - Propulsion

• Two (2) propulsion modules
• Able to instantaneously thrust in any direction and torque about any axis
• Acceleration dependent on weight of GSP payload, but designed to be up to:
  • 10 cm/s² linear
  • 30deg/s² angular
• Maximum thrust: 600mN
Interface - Human

• Touch Screen
• RGB signal lights
• Laser pointer
• Speaker
• Microphone
Interface - Software

- GSP payloads communicate to Astrobee via the Guest Science Android ROS bridge
- A JAR library will facilitate GSP payload software.
- GSP payloads subscribe to any Astrobee message.
- Advanced users may access lower level functionality.
Interface - Ground

- Easy operator interface
- Custom data message for GSP Payload
- Start, stop and send custom commands to GSP payload
Development - Simulator

- Runs actual flight software
- Simulate sensor inputs
- Runs actual GSP Payload code
Development – Payload Tester

• Main purpose to test payload port functionality and demonstrate GSP payload development path.
• External processor capable of running Linux and Core Flight Software.
• Trade study underway to select alternative attachment options, including seat track.