2nd Kibo Robot Programming Challenge

Guidebook

Version 1.3 (Released Date: May 12th, 2021)

Japan Aerospace Exploration Agency (JAXA)

List of Changes

All changes to paragraphs, tables, and figures in this document are shown below.

Release Date	Revision	Paragraph(s)	Rationale
February 8th, 2021	1.0	All	-
March 16th, 2021	1.1	3.3	Revised Game flow
April 1st, 2021	1.2	2.2.1 2.2.2	Definition of Programming Skill Round is introduced.
May 12th, 2021	1.3	2.2.1	Changed Event Schedule

Contents

1. Introduction

1.1. Purpose of Kibo-RPC

The Kibo Robot Programming Challenge (Kibo-RPC) is an educational program in which students solve various problems by moving free-flying robots (Astrobee and Int-Ball) with student's programming skills in the International Space Station (ISS). It is hoped that, by getting the opportunity to talk with professional scientists and engineers and observe the professionals work up close, they will be inspired to develop their own educational and professional goals to a high level. Participants will have the chance to learn cutting-edge methodologies and to hone their skills in science, technology, engineering, and mathematics through this program.

This program is hosted by the Japan Aerospace Exploration Agency (JAXA) in cooperation with the National Aeronautics and Space Administration (NASA).

1.2. Educational Objective for 2nd Kibo-RPC

In 2nd Kibo-RPC, experience of moving an actual robot after simulating teaches students that a simulation can only approximate the real world. Thus, participants are expected to learn techniques for creating simulation programs that perform well in the real world while considering uncertainties and error. For these reasons, students will be able to learn the necessity of controlling and correcting positions, orientating a free-flying robot and how to perform assigned tasks in the onboard environment through simulation trials.

1.3. Previous Kibo-RPC

The 1st Kibo-RPC was held between October 2019 to October 2020. In June 2020, a Preliminary Round was held in each country/region to select a representative team. The representative teams from 7 countries/regions competed against each other with their program on-orbit on October 8th, 2020. Bangladeshi students joined the competition as observers.

Table 1.3-1 Number of participating teams

Australia	14
Indonesia	37
Japan	12
Singapore	3
Taiwan	58
Thailand	151
UAE	38
Total	**313 teams** **1168 people**

- 1st Kibo-RPC Website
https://jaxa.krpc.jp/
- YouTube
https://youtu.be/UhTz_ukm1cE
- KUOA News
https://iss.jaxa.jp/en/kuoa/news/kibo-rpc_pre.html
https://iss.jaxa.jp/en/kuoa/news/kibo-rpc_final.html

*Students from Bangladesh participated as observers adding up to **361** teams, and **1340** students participated.

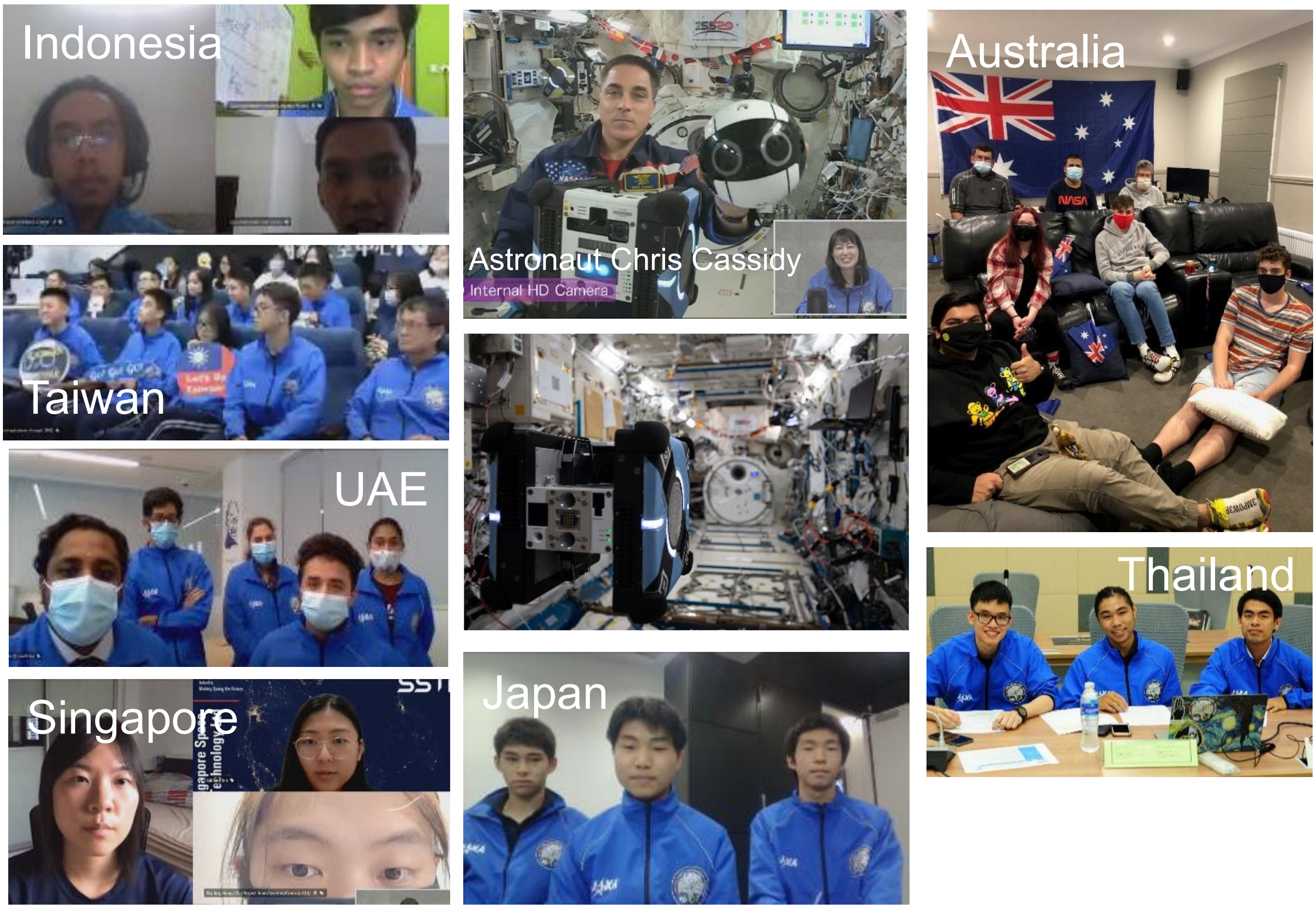


Figure 1.3 Photos in the 1st Kibo-RPC

1.4. Introduction of robots in ISS

Robots such as Astrobee and Int-Ball are on the ISS. Participants create their own program for moving Astrobee to designated locations in the Kibo-RPC.

What is Astrobee?

Figure 1.4-1 Astrobee

Astrobee, NASA's new free-flying robotic system, will help astronauts reduce the time they spend on routine duties, leaving them to focus more on the things that only humans can do. Working autonomously or via remote control by astronauts, flight controllers, or researchers on the ground, the robots can perform tasks such as taking inventory, documenting experiments, or moving small items or cargo throughout the station.
(https://www.nasa.gov/astrobee)

What is Int-Ball?

Figure 1.4-2 Int-Ball

Int-Ball is a free-flying camera robot aiming to reduce crew time ultimately to zero for routine video-shooting tasks by crew in the ISS/Kibo. Similar to current consumer-grade cameras, Int-Ball works closely with onboard crew to provide flexible views for ground operators. Int-Ball is perhaps the first human-friendly camera robot in space.
(http://iss.jaxa.jp/en/kiboexp/news/171214_int_ball_en.html)

2. Event Information

2.1. Participation

✓ Please refer to the Entry Description regarding the participation.

2.2. Event Plan

2.2.1. Event Schedule

Feb 2021

Call for Participation

- We might have a guidance session for applicants.

Self-Learning

- Participants need to access to the Github repository provided by NASA to learn about the programming of the space robot (Astrobee).

Program Development

- Release of the web simulation environment is planned in April 2021.
- Participants create program for JAXA's simulation environment.

Jun 2021

Preliminary Round End of June to Early July

- It will be held in each country/region by domestic space agency to select the national representative.
- The winning team can go on to the Programming Skills Round.

Programming Skills Round Middle of July

(Specific date and locations are still under coordination)

- This round will be held to award the best programming skilled teams out of the winners of Preliminary Round.
- Also, finalists for the Final Round are selected. The number of the finalists is not decided yet

Program Refine

- Finalists refine program for ISS final round.
- Release of web simulation for Final Round is planned in end of July 2021.

Sep 2021

Final Round Around September 2021

- It will be hosted by JAXA at Tsukuba Space Center.
- It is the event connected with ISS and broadcasted on YouTube.

Table 2.2.1 Event Schedule

2.2.2. Event Details

(1) Preliminary Round

To select representatives from each country/region, participants must take part in a preliminary competition by using the simulator organized by their country/region POC.

- Teams submit a program for JAXA's simulation before this round.
- Teams compete against each other by speed and accuracy of a mission in simulations.
- One winning team is selected as a representative who can participate in the qualification round or the final round.
- The competition is judged based on JAXA's scoring factors and game rules.
- Details of game rules are in the Kibo RPC Rule Book.

Detailed information, such as venue and schedule, will be announced by the POC of each country/region.

(2) Programming Skills Round

The teams who are selected from the Preliminary Round are nominated for the Best Programming Skills Award. The skills are awarded based on the results of in JAXA's simulation environment, before they are run in a real robot on ISS.

Also in this round, top teams can move to the final round. Since the number of the finalists depends on the ISS crew's schedule, the plans will be announced later.

- Teams don't need to submit a program again that was developed for the preliminary round.
- Teams compete with representatives from other country/regions on a simulator.
- A number of finalists are selected for the Final Round.
- The scoring factors and game rules are the same as the Preliminary Round, but random factors in the simulator will be changed.
- Details of game rules are written in the Kibo RPC Rule Book.

Information related to the event is announced by Kibo-RPC secretariat.

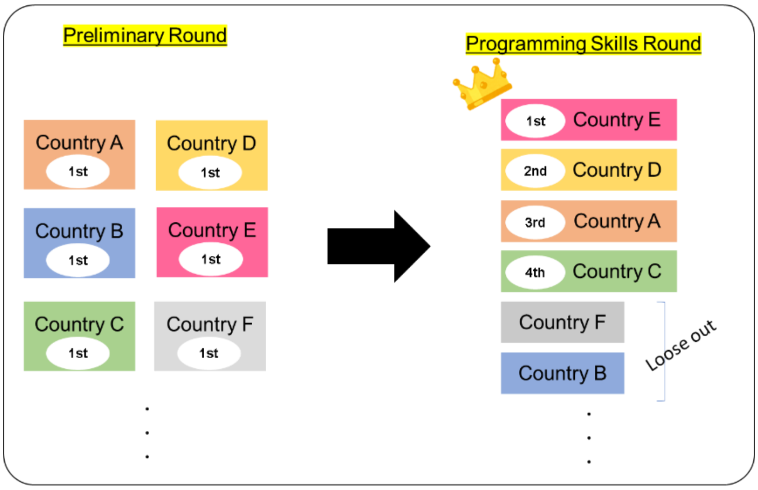


Table 2.2.2 Example for Finalist Selection at Programming Skills Round

(3) Final Round

Finalists selected at the Programming Skill Round can reach the Final Round.

- Each team's program developed for JAXA's simulation environment is uplinked to Astrobee on the ISS.
- Teams need to submit their program to Kibo-RPC secretariat in advance.
- Teams can modify the program used in previous rounds before submitting.
- The competition will be judged based on JAXA's scoring factors and game rules.

Detailed information such as program submission date will be announced around the Preliminary Round.

Date: Around September 2021

Venue: Tsukuba Space Center (TKSC) (https://global.jaxa.jp/about/centers/tksc/index.html)

The final round will be streamed live in each Kibo-RPC country/region for participants who cannot travel to Japan.

Ask the POC in each country/region about travel expenses and accommodation fees and organization.

2.2.3. Release of Web site and Simulation

The web site including web-based simulation will be prepared in phases. Until teams are issued with new identification, refer to the 1st Kibo-RPC Web Site. (https://jaxa.krpc.jp/1st/index.html)

Table.2.2.3 Release Schedule

Event	Date
Programming Manual	Middle of March, 2021
Open Web Site for Participants	April 1st, 2021
Simulation	April 1st, 2021
Update for Final Round	July ,2021

3. Game Description

3.1. Scenario

On October 2020, the International Space Station (ISS) and its Kibo Module, was damaged as it was hit by a meteor shower and this caused an air leakage. Kudos to young programmers in Asia-Pacific Region who repaired the leakage in the nick of time. Life went back to normal.

Now in 2021, the repair turned out to be insufficient and the air leakage recurred! The fate of the astronauts is once again in danger!

Since astronauts have to stay in a safe zone, they cannot go close to the leaking area! It is time for Astrobee to go for the repair mission again. The location of the leak hole has been identified based on the previous mission, and programmers are needed who can manipulate Astrobee with quick and precise control.

The flight controllers detected with various instruments on the ISS that in the vicinity of the leak, there is a string air flow disturbance, making it impossible to approach in a perpendicular path. Debris in the Kibo Module is preventing a short and straight trajectory of the Astrobee. Moreover, the direction of the hole was found NOT to be perpendicular to the surface of the wall. This time, a complete repair is needed to prevent a recurrence. The mission is to have no more air leaks! Use Astrobee to detect the precise direction of the hole and weld it accurately and completely using Astrobee's laser.

Astronauts have started their emergency evacuation procedures. When you succeed in your mission, you need to report to the astronauts to cancel their procedures and ask them to inspect the repair. You need a clear and strong message to get the attention of the astronauts.

Emergency mission to the student programmers in Asia-Pacific Region.

!!Mission!!

Complete air leak repair and report to astronauts!

*This story is fiction.

3.2. Game Overview

Teams create a program to move Astrobee from docking station to the specified places and point the laser a target. After that, teams report “mission complete” to an astronaut.
The score will be calculated by the combination of the accuracy of laser pointing on the target, the elapsed time. (Calculation method will be updated.)

3.3. Game flow

① Move the Astrobee from the dock station to Point-A

② Read a QR code at Point-A and get the information of a point where target position can be in sight (Point-A') and Keep Out Zone (KOZ)*[1] in front of the target.

③ Move to Point-A'

④ Irradiate the center of the target with a laser by utilizing information of relative position from the AR tags while avoiding the obstacle (KOZ).

⑤ If you wish to fine adjust the position for better targeting, you can program to automatic re-try within the time limit.

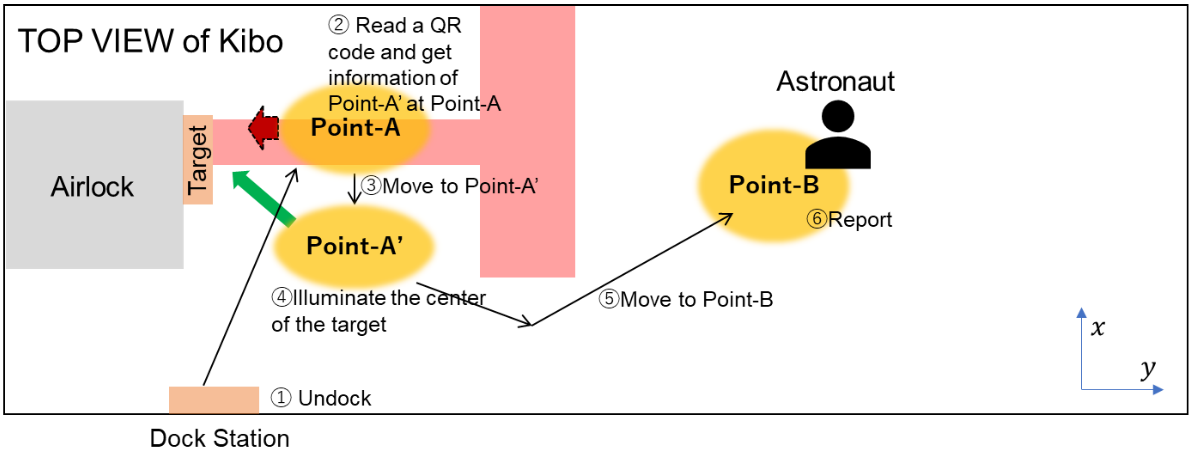
⑥ Move to Point-B in backward motion with facing the Airlock direction.

⑦ Report "Mission Complete" to Astronaut. *[2]

*[1]Keep Out Zone(KOZ)･･･

The area Astrobee cannot move into. If Astrobee attempts to go there, it is rejected.

*[2]Time limit is 5 minutes. (Mission finishes with the mission complete command.)



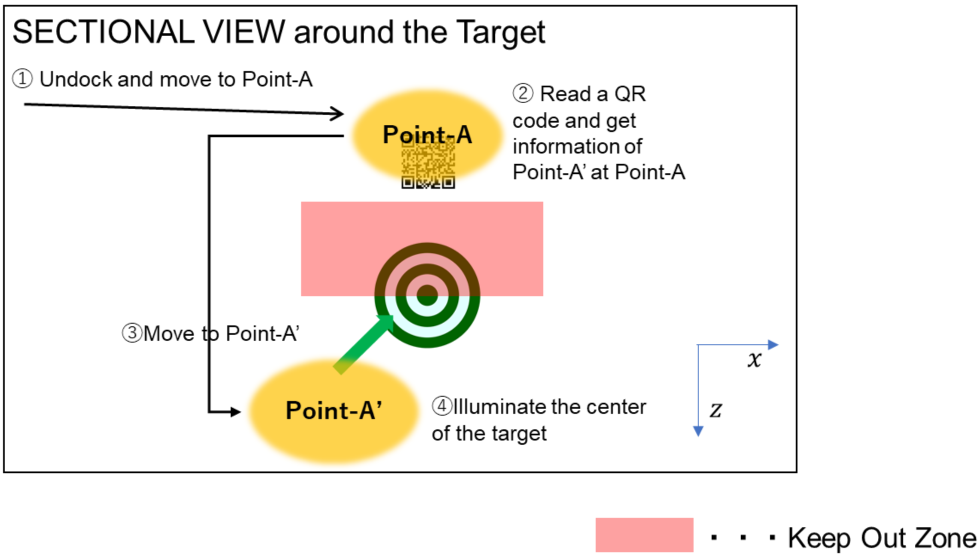


Figure 3.3-1 Game scenario

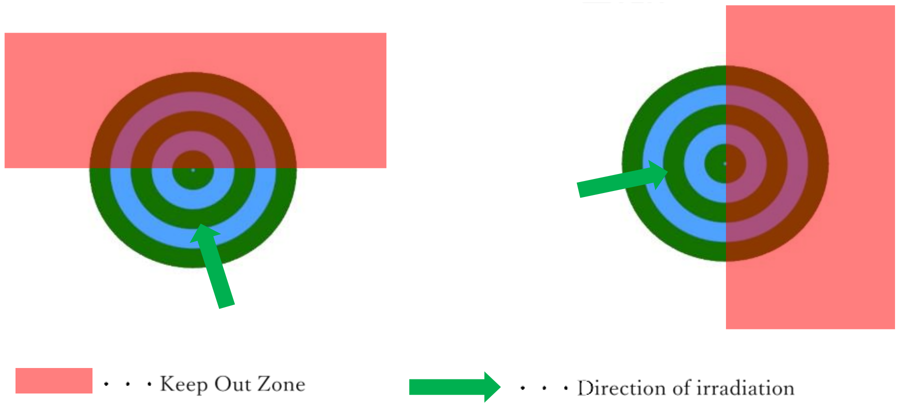
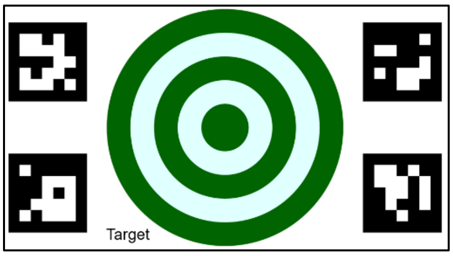


Figure 3.3-2 Example pattern of the KOZ in front of the Airlock



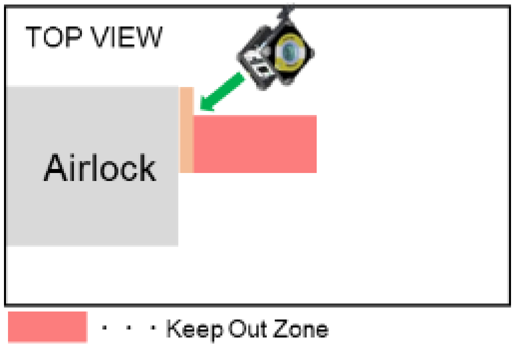


Figure 3.3-3Target

Figure 3.3-4 Irradiation target from diagonal direction

3.4. Released information in the future

The followings information will be released in a Kibo RPC Rule Book.

1. Coordinates and orientation when docking
2. Coordinates and orientation of Point-A
3. Coordinates and orientation of Point-B
4. Coordinates of KOZ in front of the target
5. Coordinates of KOZ between Point-A to Point-B
6. Positional relation between QR code, AR tag and the target
7. Size of the AR tag
8. Size of the QR code
9. Size of the target

3.5. Evaluation standard

Your program is assessed based on the accuracy of laser irradiation and the elapsed time of completing the mission. 2 elements are evaluated, and your score is calculated between 0 to 100.

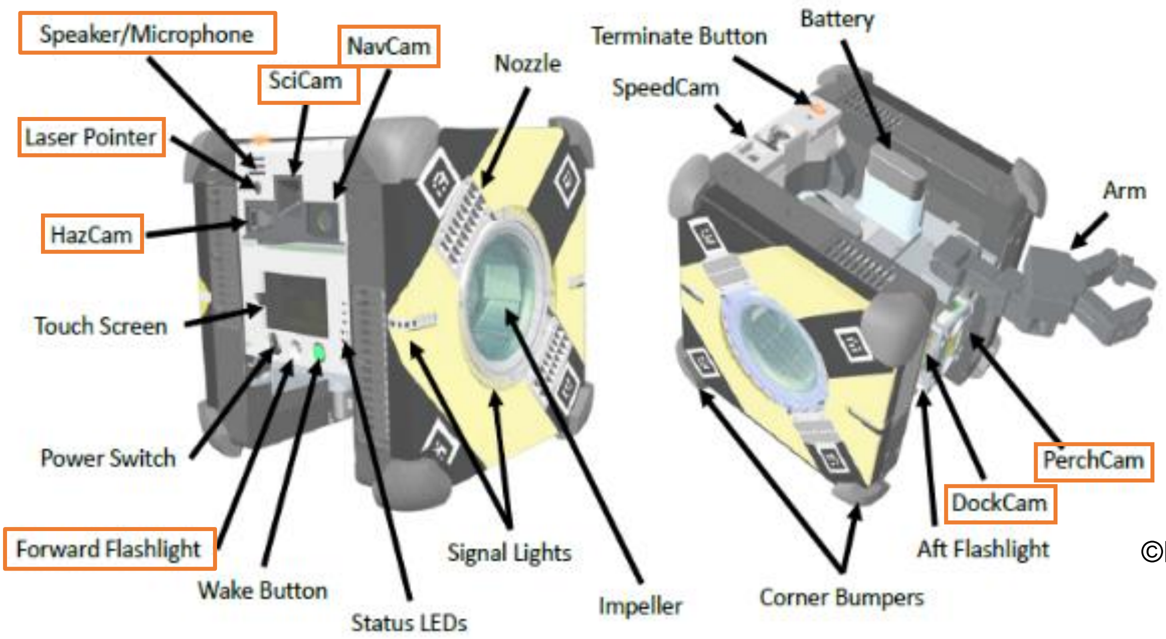
Accuracy	Distance between the center of the target and hit point.
Time	Time from undocking command to mission complete command.

3.6. Tips for Astrobee Characteristics

The tips for Astrobee characteristics will be provided in this section to be considered for successful runs.

3.6.1. Rendering of Astrobee

Figure 3.6.1-1 indicate Astrobee is equipped with some external hardware components. The hardware surrounded with one-foot cube (about 32 cm wide) are used in the simulator and ISS final.



SciCam	A color camera for taking a video (The participants cannot utilize. In final round, Sci Cam is used to take color videos, and acts as Astrobee's eye.)
DockCam	A monochrome camera for docking to the docking station (In the simulator, it takes videos of rear.)
PerchCam	A monochrome camera for grabbing a handrail (You can create a program with this camera, if needed.)
Laser Pointer	Irradiating the target (It has a distance from NavCam. Be careful when you create a program of image data processing. The detail of distance, please refer to Programming Manual clause 5.4)
Flashlight	Use this when read QR code
Speaker	Report "mission complete" to an astronaut with your voice

Table 3.6.1-1 Rendering of Astrobee

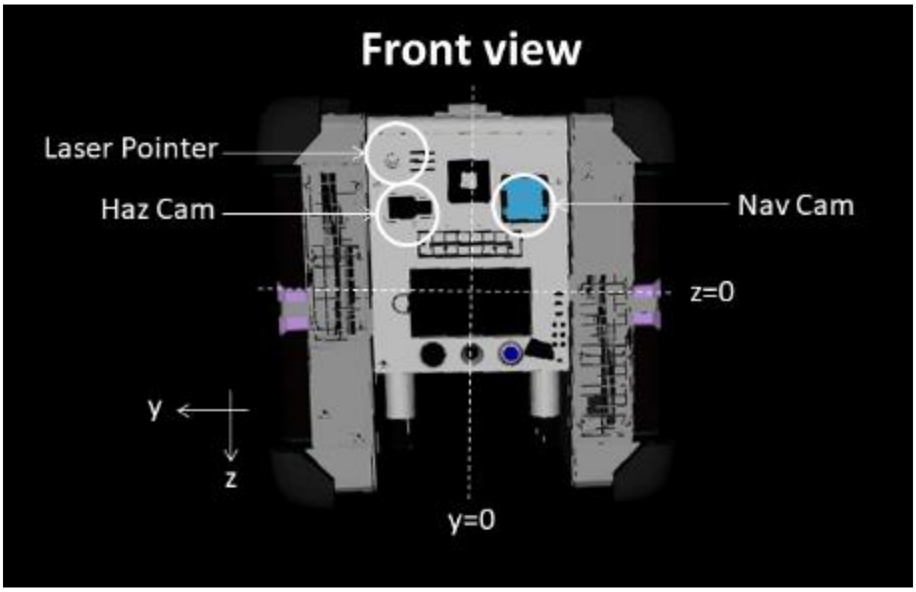


Figure 3.6.1-2 Astrobee Front View

Table 3.6.1-1 Distances from center point

	x[m]	y[m]	z[m]
Nav Cam	0.1177	-0.0422	-0.0826
Haz Cam	0.1328	0.0362	-0.0826
Laser Pointer	0.1302	0.0572	-0.1111

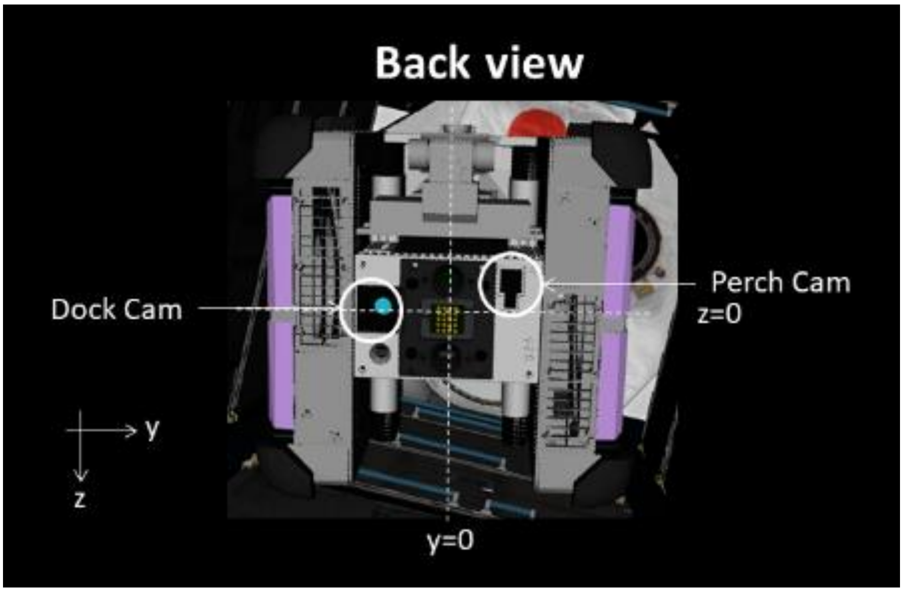


Figure 3.6.1-3 Astrobee Back View

Table 3.6.1-2 Distances from center point

	x[m]	y[m]	z[m]
Dock Cam	-0.1061	-0.054	-0.0064
Perch Cam	-0.1331	0.0509	-0.0166

3.6.2. Specification of Astrobee

* Mass: 10kg (Installed only two of the four batteries)
* Maximum Velocity: 0.5 m/s
* Maximum Thrust (X axis): 0.6 N
* Minimum moving distance: 0.05 m.
* Minimum rotating angle: 7.5 degrees.
* If the Astrobee detects the actual obstacles in front, Astrobee will automatically stop and then maintain its position and orientation. The moving path of Astrobee in process is also discarded.

3.6.3. References of Astrobee

- GitHub-1 (https://github.com/nasa/astrobee)
- GitHub-2 (https://github.com/nasa/astrobee_android)
- Website of Astrobee (https://www.nasa.gov/astrobee)