3rd Kibo Robot Programming Challenge
Guidebook

Version 1.0 (Released Date: February 18th, 2022)

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 18th, 2021	1.0	All	-

Contents

1. Introduction 1
1.1. Purpose of Kibo-RPC 1
1.2. Educational Objective for 3rd Kibo-RPC 1
1.3. 1st Kibo-RPC 2
1.4. 2nd Kibo-RPC 3
1.5. Introduction of Robots in ISS 4
2. Event Information 5
2.1. Participation 5
2.2. Event Plan 5
2.2.1. Event Schedule 5
2.2.2. Event Details 6
(1) Preliminary Round 6
(2) Final Round 6
2.2.3. Release of Web Site and Simulation 6
3. Game Description 8
3.1. Scenario 8
3.2. Game Overview 8
3.3. Game flow 9
3.4. Released information in the future 10
3.5. Evaluation standard 10
3.6. Tips for Astrobee Characteristics 10
3.6.1. Rendering of Astrobee 10
3.6.2. Specification of Astrobee 12
3.6.3. References of Astrobee 12

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) using their programming skills in the Japanese Experiment Module "Kibo" of the International Space Station (ISS). By getting the opportunity to talk with professional scientists and engineers and observe their work up close, it is hoped the students 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 3rd Kibo-RPC

In the 3rd Kibo-RPC, students learn that only the experience of moving an actual robot in 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 errors. For these reasons, students will be able to learn the necessity of controlling and correcting positions, attitude a free-flying robot and how to perform assigned tasks in the onboard environment through simulation trials.

1.3. 1st Kibo-RPC

The 1st Kibo-RPC was held between October 2019 and 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 in the ISS (on-orbit) on October 8th, 2020. In addition, a team of 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://humans-in-space.jaxa.jp/krpc/1st/index.html
- 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

*There were **48** teams and **172** students from Bangladesh as observers.

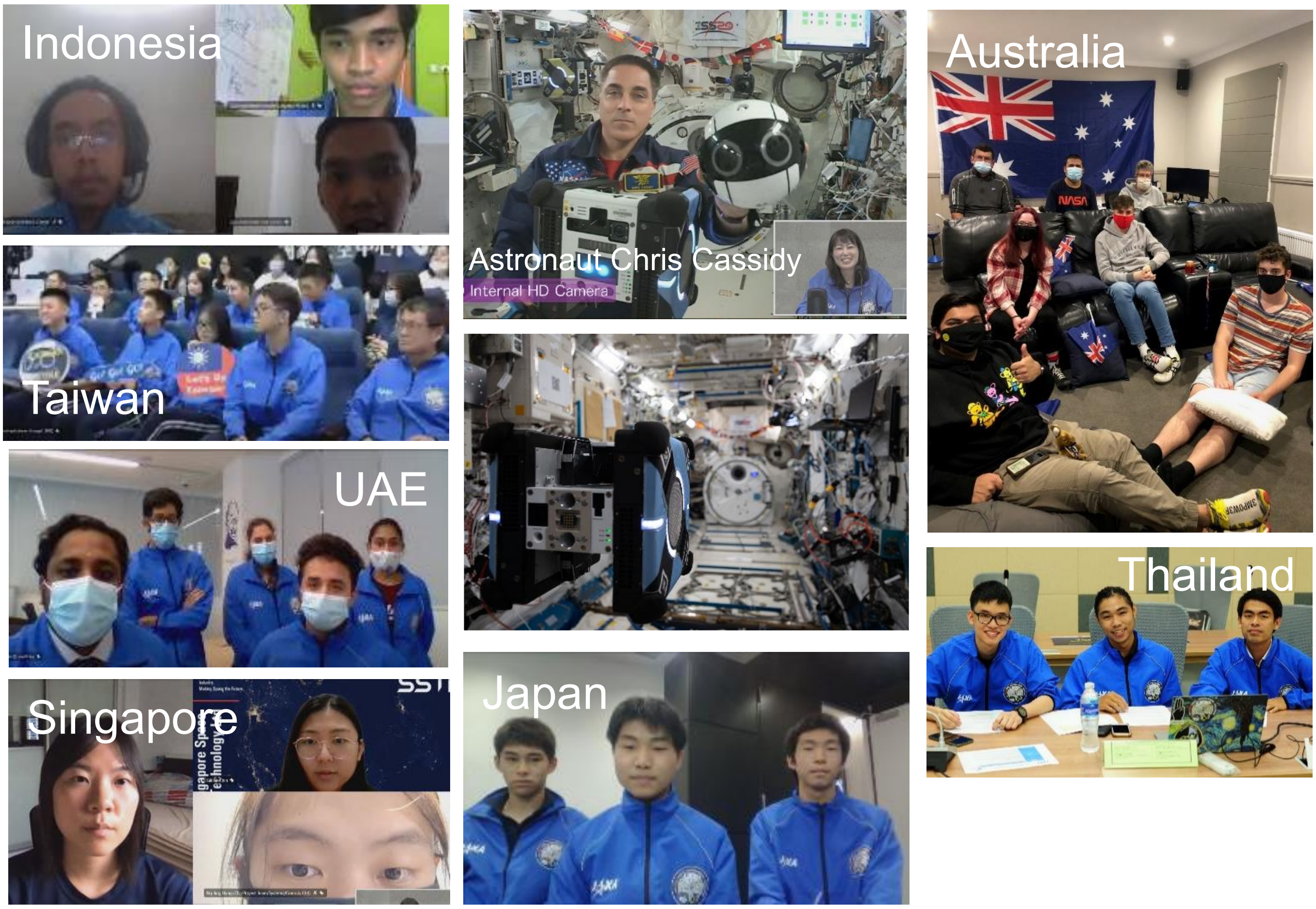


Figure 1.3 Photos in the 1st Kibo-RPC

1.4. 2nd Kibo-RPC

The 2nd Kibo-RPC was held between February 2021 and October 2021. Preliminary Round was held in each country/region in July 2021, and the representative teams participated in the Programming Skills Round to rank themselves in the simulation environment. The teams flew in the Final Round in this ranking order. On October 24, 2021, an in-orbit final event was held with the result videos of the programs created by representative teams from 9 countries/regions, which were uploaded to the Astrobee in the ISS (in-orbit) where they competed against each other. Teams from Nepal and New Zealand participated as observers.

Table 1.4-1 Number of participating teams

Australia	15	**New Zealand**	4
Bangladesh	22	**Singapore**	5
Indonesia	13	**Taiwan**	18
Japan	19	**Thailand**	176
Malaysia	12	**Vietnam**	1
Nepal	1	**Total**	**286 teams** **905 people**

- 2nd Kibo-RPC Website
  https://humans-in-space.jaxa.jp/krpc/2nd/index.html
- YouTube
  https://www.youtube.com/watch?v=eDXf1ISUBmA&t=20s
- KUOA News
  https://humans-in-space.jaxa.jp/en/biz-lab/news/detail/001689.html

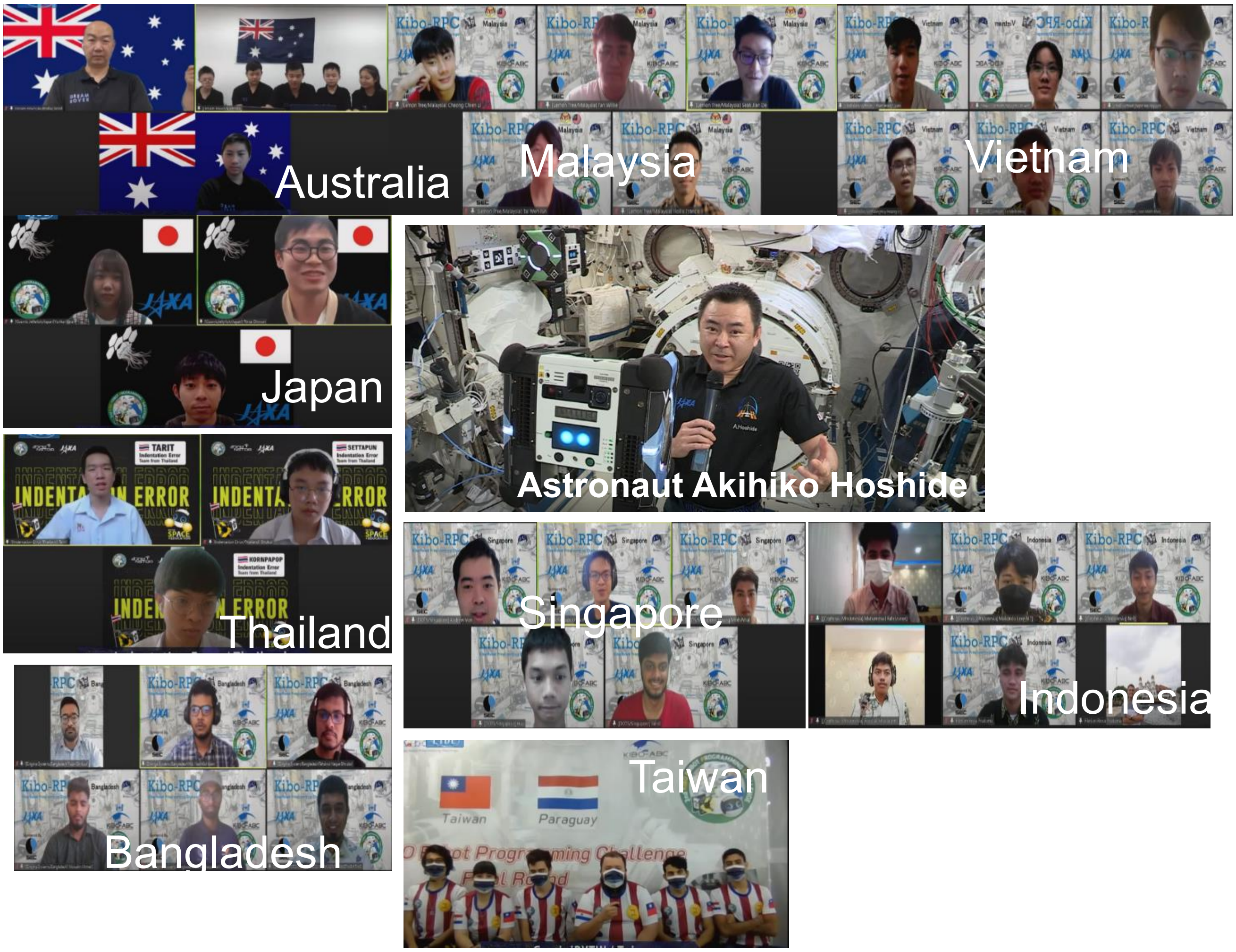


Figure 1.4 Photos in the 2nd Kibo-RPC

1.5. 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.

*Int-Ball will not be used in the competition.

What is Astrobee?

Figure 1.5-1 Astrobee

Astrobee is NASA's new free-flying robotic system that 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 1.5-2 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 2022

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).
- A video of the program development procedure for beginners is available on the website for self-study. (To be released as soon as it is ready.)

Program Development
- Release of the web simulation environment is planned in April 2022.
- Participants create program for JAXA's simulation environment.

July 2022

Preliminary Round Early July
- It will be held in each country/region by domestic space agency to select the national representative.

Program Refine
- Finalists refine program for ISS final round.
- Release of web simulation for Final Round is planned in end of July 2022.

Sep 2022

Final Round Around September 2022
- 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 are required to develop and submit a program for JAXA's simulation by the specified date.
- Teams compete against each other by speed and accuracy of a mission in simulations.
- One winning team is selected as a representative to participate in the Final Round.

  However, due to the limitation of the number of teams to conduct simulations in orbit, the final number may be further reduced from the representatives of each country based on their performance in the Preliminary Round. Since the number of finalist teams depends on the ISS crew's schedule, the plans will be announced later.
- The competition is judged based on JAXA's scoring factors and game rules.
- Details of the 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.

If the Worldwide team (see the Entry Description) loses in the Preliminary Round of each country/region, they can participate in the Bonus Round conducted by the Kibo-RPC Secretariat. The winning team can also advance to the Final Round. Detailed information will be announced by the Kibo-RPC secretariat at a later date.

(2) Final Round

Finalists selected during the Preliminary Round can reach the Final Round.

- Each team in the final round is required to develop a program in JAXA's simulation environment and submit it to the Kibo-RPC secretariat by the designated date.
- The submitted programs will be checked by the Kibo-RPC secretariat and the NASA Astrobee development team, and will be uplinked/installed on the Astrobee on the ISS.
- The competition will be judged based on JAXA's scoring factors and game rules.

Detailed information such as program submission date will be announced later.

Date: Around September 2022

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

The Final Round will be streamed live on YouTube.

The details of how the event will be held will be announced as soon as they are determined.

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 previous Kibo-RPC Web Site.

(1st Kibo-RPC ：https://humans-in-space.jaxa.jp/krpc/1st/index.html、

2nd Kibo-RPC：https://humans-in-space.jaxa.jp/krpc/2nd/index.html)

Table.2.2.3 Release Schedule

Event	Date
Programming Manual	April 1st, 2022
Rule Book	April 1st, 2022
Simulation	April 1st, 2022
Update for Final Round	July ,2022

3. Game Description

3.1. Scenario

The ISS air leak caused by a meteor impact in 2020 was fixed thanks to the efforts of young programmers in Asia.

In 2021, the air leak reappeared, and the repair mission was relaunched. Unfortunately, this time it was only a temporary fix.

In 2022, the ISS was hit by space debris, which has become a problem in recent years.The impact caused another air leak, resulting in a drop in the ISS's air pressure and scattering objects inside Kibo.

A telemetry check was conducted by flight controllers, which found the speed of the pressure drop faster than expected, suggesting that there might be two holes.

One of the holes was fortunately identified by video analysis of the onboard camera. Since the hole was still small and the material was quickly melted by the heat of a laser, it seems that you can weld by simply hitting it in the vicinity of the hole. It is also fortunate that the location is easily accessible by Astrobee.

The other hole was more complex, and it seemed to be in the same location where we had done repairs the last time. The impact of the debris may cause the hole to open wider, but the only way to confirm is for the Astrobee to go in front of the hole and observe it. If the repaired area is the cause, we must also consider the deterioration of the material since it has been welded many times. Irradiation to the exact location is desirable.

As before, the astronauts have been evacuated to a safe location and only the Astrobee can carry out the mission. Once the scattered obstacles have been successfully avoided and the air leak has been stopped, please reassure astronauts by reporting the repairs have been completed.

This time, we need to repair the air leak wholly and accurately.

Will this third time be the charm?

!!Mission!!

Young programmers, save the ISS from facing another crisis!

*This story is fiction.

3.2. Game Overview

Teams create a program to move Astrobee from the start position to two specified places and point the laser at targets. After that, teams report “mission complete” to an astronaut.

The score will be calculated by a combination of the accuracy of laser pointing on the target and the elapsed time.

3.3. Game flow

① Move the Astrobee from the start position to the vicinity of Point1.

② Irradiate the laser in the area inside the green square of Target1 using the surrounding AR tags.

③ Move the Astrobee to the vicinity of Point2 while avoiding the Keep Out Zone (KOZ)*1, which looks like an obstacle.

④ Irradiate the center of Target2 with a laser by utilizing information of the relative position from the AR tags.

If you wish to adjust the position for better targeting, you can program for an automatic re-try if it is within the time limit.

⑤ Avoid KOZ and move to Goal in backward motion while facing the Airlock direction.

⑥ Report "Mission Complete" to Astronaut using the Astrobee's functions. *2*3

*1Keep Out Zone(KOZ)･･･

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

*2Time limit is 5 minutes. (Mission finishes with the mission complete command, it does not include the time to report to Astronauts)

*3Implementation methods will be presented in the Rule Book.

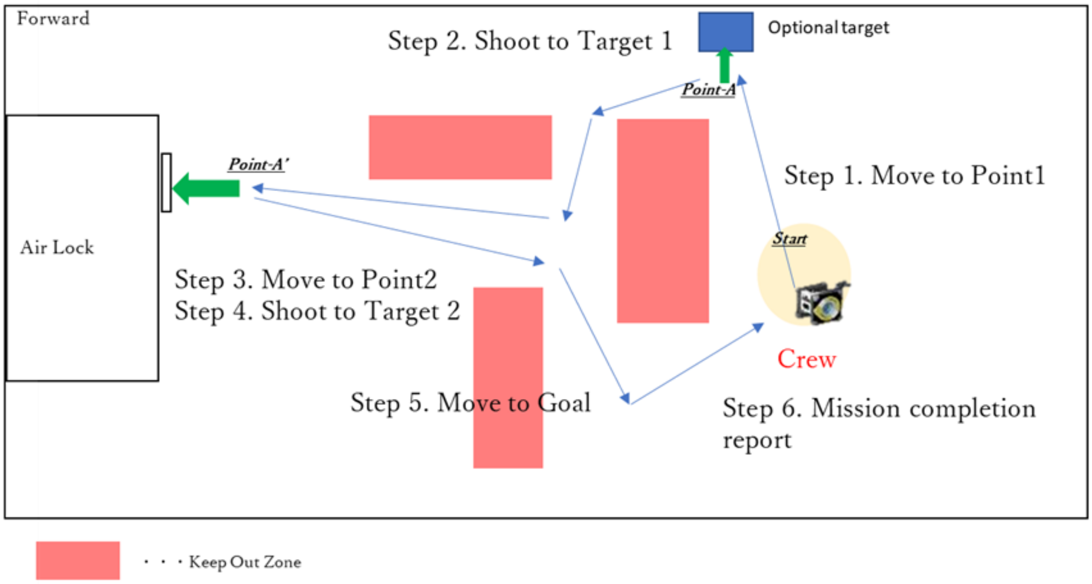


Figure 3.3-1 Game scenario

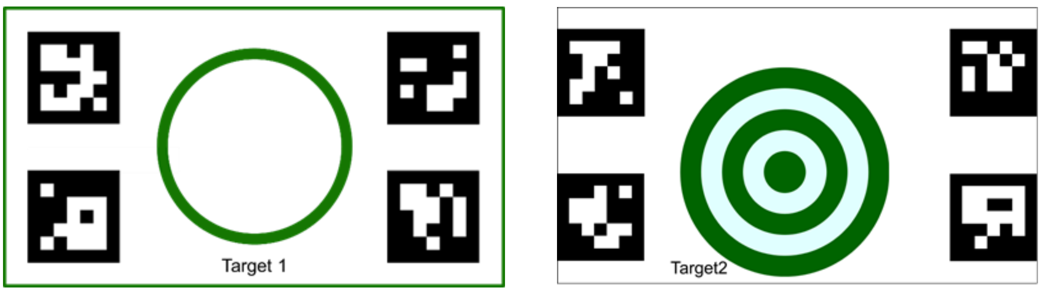


Figure 3.3-2Target (Left: Target1, Right: Target2)

3.4. Released information in the future

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

1. Coordinates and orientation when starting
2. Coordinates and orientation of Point1
3. Coordinates and orientation of Point2
4. Coordinates of KOZ
5. Positional relation between AR tags
6. Size of the AR tag
7. Size of the targets

3.5. Evaluation standard

Your program is assessed based on the accuracy of laser irradiation and the elapsed time of completing the mission. Two elements are evaluated, and your score is calculated. For the 3rd Kibo-RPC, scoring will be done in such a way that those who can complete the mission more accurately will be given a higher rating. The details will be presented in the Rule Book.

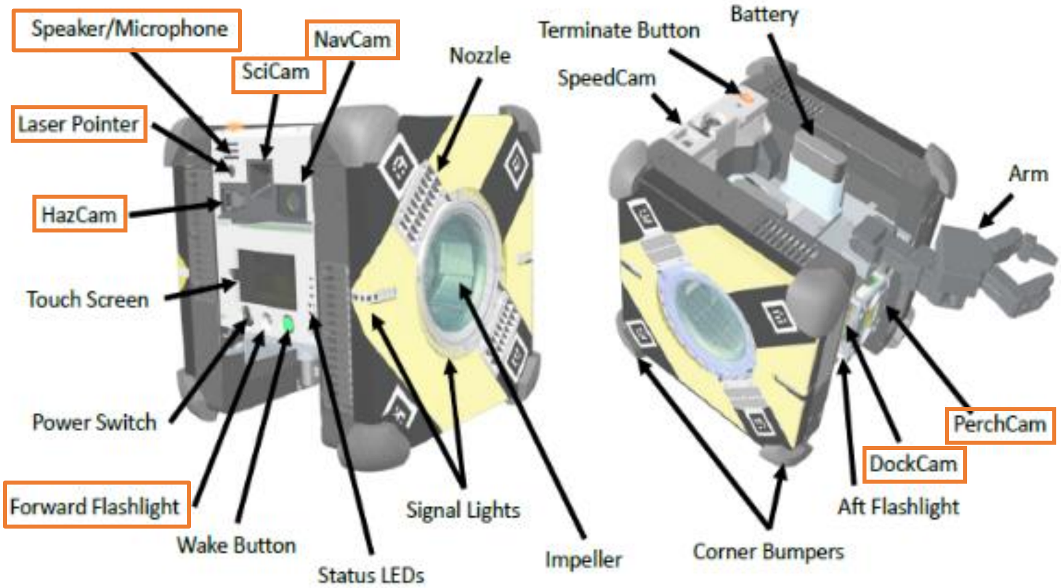
Accuracy	Distance between the center of the target and hit point.
Time	Time from undocking command to mission complete command.

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 indicates 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 Round.



©NASA

Name	Explanation
HazCam	A monochrome camera for detecting obstacles within 30 cm
NavCam	A monochrome camera for image data processing and taking a photo after sending finish command
SciCam	A color camera for taking a video (The participants cannot utilize this. In the final round SciCam 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 the 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. For details of distance, please refer to Programming Manual clause 5.4)
Flashlight	Use this when reading QR code
Speaker	Report "mission complete" to an astronaut by your voice

Table 3.6.1-1 Rendering of Astrobee

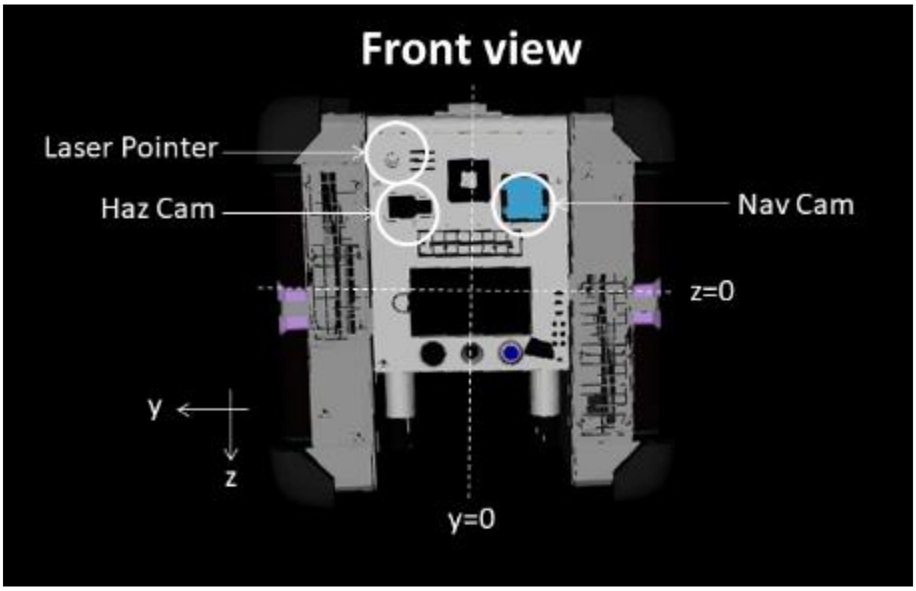


Figure 3.6.1-2 Astrobee Front View

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

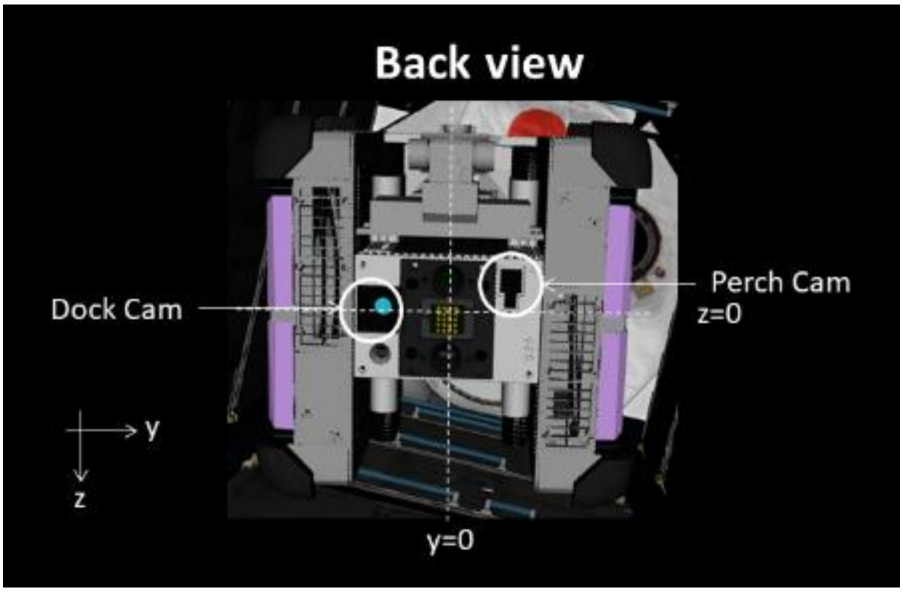


Figure 3.6.1-3 Astrobee Back View

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, it will automatically stop and then maintain its position and orientation. The moving path it is on will also be 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)