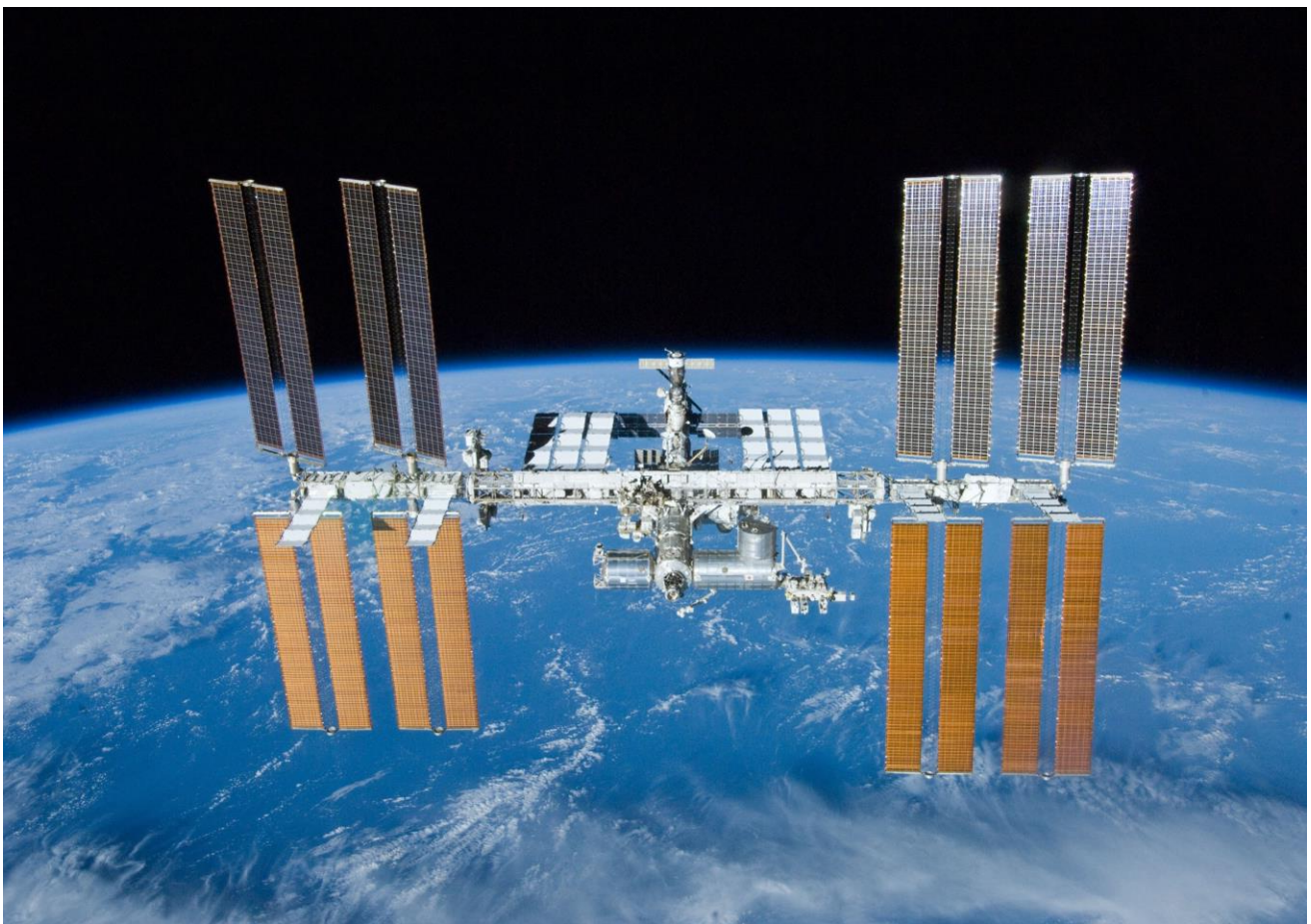JAXA

1st Kibo Robot Programming Challenge
Guidebook

Version 2.10 (Released Date: May 25th, 2020)

Japan Aerospace Exploration Agency (JAXA)

General Point of Contact: Z-KRPC@ml.jaxa.jp

List of Changes

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

Release Date	Revision	Paragraph(s)	Rationale
October 11th, 2019	1.0	All	-
November 20th, 2019	2.0	All	Change of document style and expression.
		1.4.2(3)	Correction of erroneous description
		1.7	Addition and clarification of condition
December 25th, 2019	2.1	2.3.1(2), (4), (5) 2.3.3(2), (4), (5)	Addition and clarification of condition
		2.4	Addition
January 29th, 2020	2.2	1.4.1	Add the member county/region of Kibo-ABC and Kibo-RPC
February 26th, 2020	2.3	1.4.1	Add the member county/region of Kibo-RPC
		1.6.2	Modify the schedule of Preliminary Round
March 11th, 2020	2.4	1.5	Change of the deadline
March 24th, 2020	2.5	1.4.1	Add the member county/region of Kibo-ABC and Kibo-RPC
April 13th, 2020	2.6	1.4.1	Add the member county/region of Kibo-ABC
April 20th, 2020	2.7	1.5	Change of the deadline
May 1st, 2020	2.8	1.4.1	Delete document
May 18th, 2020	2.9	2.4.2	Modify the Specification of Astrobee
May 25th, 2020	2.10	2.3.1 2.3.3	Corrected an omission in the description of time limit constraints.

1. Contents

1. Event Information

1.1. Purpose of Kibo-RPC

The Kibo Robot Programming Challenge (Kibo-RPC) is an educational program in which students solve various problems by programming free-flying robots (Astrobee and Int-Ball) in the International Space Station (ISS). It is hoped that, by providing these students with the opportunity to work with professional scientists and engineers, 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. The Kibo-RPC will also expand international exchange by encouraging students to interact with other participants from around the world.

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 1st Kibo-RPC

Engineering teaches us 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 despite uncertainties and within margins of error.

Students will learn the necessity of controlling and correcting positions and orientation of a free-flying robot and how to perform assigned tasks in the onboard environment through simulation trials.

1.3. Collaboration with Astrobee and Int-Ball

In the 1st Kibo-RPC, participants will create a program to move Astrobee to designated locations. Int-Ball will serve as a camera robot to achieve tasks assigned to participants. From the 2nd Kibo-RPC, Int-Ball is expected to actively solve assigned tasks.

What is Astrobee?

Figure 1.3-1 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 1.3-2 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)

1.4. Participation

1.4.1. Entry Qualification

Students up to graduate school in a Kibo-ABC member country/region*1 expressing the participation in the Kibo-RPC *2 can apply for the competition. Foreign students and non-citizens may also take part.

Special case: Non-students may also be eligible to compete if they are not professional programmers. They will need to submit necessary identification and consult with the Kibo-RPC point of contact (POC) in the participating country/region.

*1 Kibo-ABC member countries/regions:

Australia, Bangladesh, Indonesia, Japan, Malaysia, New Zealand, Philippines, Rep. of Korea, Singapore, Taiwan, Thailand, United Arab Emirates, Vietnam.

Kibo-ABC URL: http://iss.jaxa.jp/en/kuoa/kibo_abc/

*2 Kibo-RPC participating countries/regions (As of March 2020):

Australia (ASA and OGL), Indonesia (LAPAN), Japan (JAXA), Singapore (SSTA), Taiwan (NSPO), Thailand (NSTDA), United Arab Emirates (UAESA)

1.4.2. Participating Teams

(1) Team Structure

- A team is comprised of 3 or more members.
- Students CANNOT join more than one team.

(2) Team Leader

- Each team must have a leader who is responsible for team management. A student in a team can be the leader.
- The team leader has the following responsibilities:
  - ✓ Manage the progress of the team
  - ✓ Submit applications and programs
  - ✓ Communicate and share information with the POC in its country/region.

(3) Other conditions

- The following skills and knowledge are recommended but not mandatory:
  - ✓ Android programming and image processing
  - ✓ College-level knowledge of physics or mathematics
- Teams should read paragraph 1.7 especially carefully.

1.5. Application

(1) Fill out the application form posted on the Kibo-RPC Web (https://jaxa.krpc.jp/) and submit it to the POC in its country/region.

Note:

While you are waiting for notification of acceptance, go to (https://github.com/nasa/astrobee, GitHub) provided by NASA to learn about the programming the Astrobee.

(2) The POC will notify applicants of their acceptance by the Kibo-RPC secretariat by email with instructions on how to access JAXA's simulation environment.

Deadline:

-Australia	May 24th, 2020(EST 24:00)
-Indonesia	May 7th, 2020(WIB 24:00)
-Japan	May 24th, 2020(JST 24:00)
-Singapore	May 7th, 2020(SGT 24:00)
-Taiwan	May 7th, 2020(CST 24:00)
-Thailand	April 19th, 2020(ICT 24:00)
-UAE	May 21th, 2020(GST 24:00)

1.6. Event Plan

1.6.1. Event Description

Preliminary Round: All entrants will have a preliminary competition by simulator in the country/region of application to select the team that will represent that country/region.

Qualification Round: Six countries/regions will proceed to the final round. If there are more than six participating countries/regions, all teams will compete in an elimination round until six finalists are left.

Final Round: The final round will be held in ISS/Kibo module. The six teams from the preliminary or qualification round will compete for the grand prize.
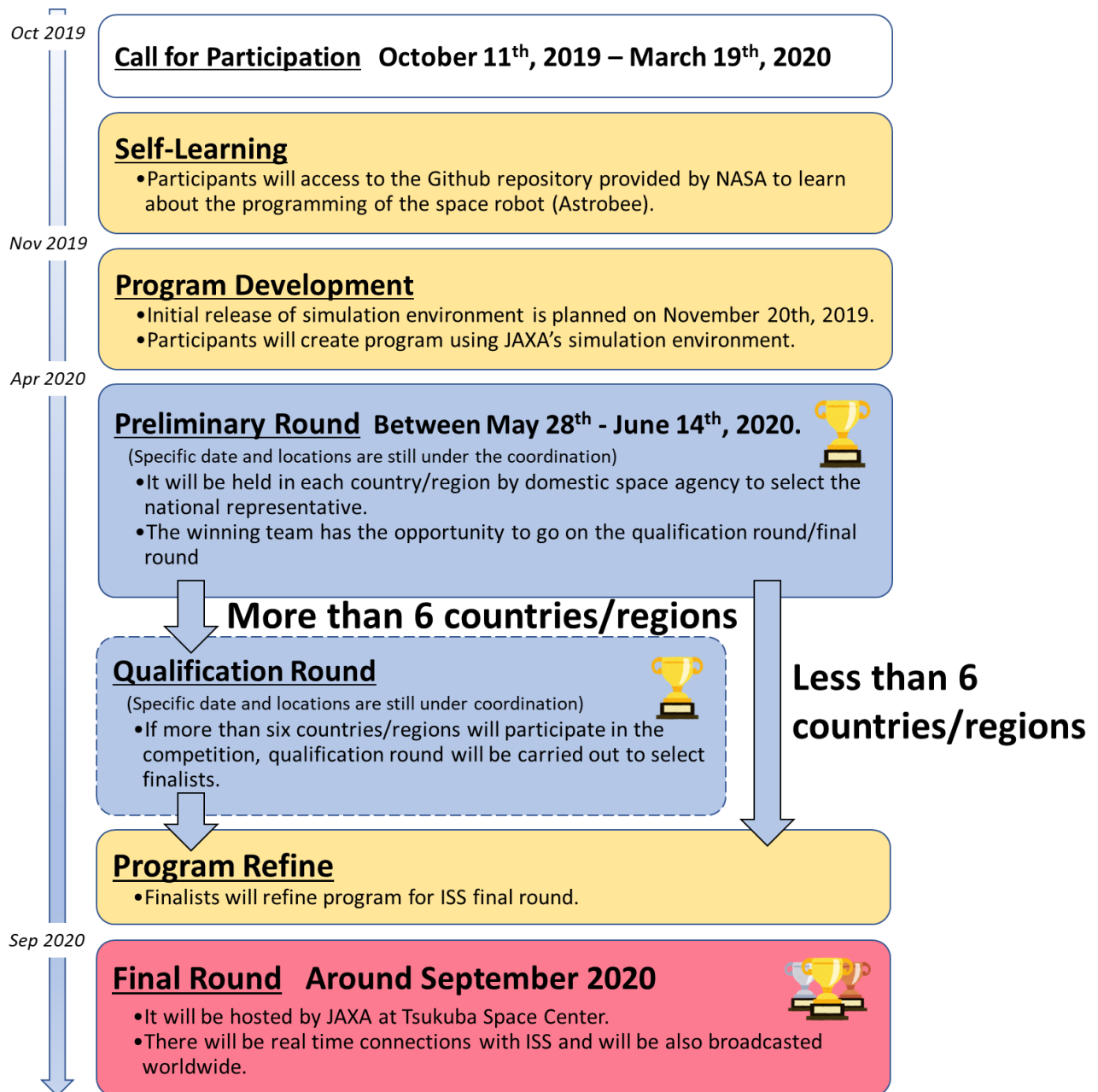
1.6.2. Event Schedule

Oct 2019

Call for Participation October 11th, 2019 – March 19th, 2020

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

Nov 2019

Program Development
- Initial release of simulation environment is planned on November 20th, 2019.
- Participants will create program using JAXA's simulation environment.

Apr 2020

Preliminary Round Between May 28th - June 14th, 2020.
(Specific date and locations are still under the coordination)
- It will be held in each country/region by domestic space agency to select the national representative.
- The winning team has the opportunity to go on the qualification round/final round

More than 6 countries/regions

Qualification Round
(Specific date and locations are still under coordination)
- If more than six countries/regions will participate in the competition, qualification round will be carried out to select finalists.

Less than 6 countries/regions

Program Refine
- Finalists will refine program for ISS final round.

Sep 2020

Final Round Around September 2020
- It will be hosted by JAXA at Tsukuba Space Center.
- There will be real time connections with ISS and will be also broadcasted worldwide.

Figure 1.6.2 Event Schedule

1.6.3. Event Details

(1) Preliminary Round

- Teams will use the program that they developed using JAXA's simulation environment before this round.
- Teams will compete against each other in simulations.
- One winning team will be selected for each country/region to participate in the qualification round/the final round.
- The competition will be judged based on JAXA's scoring factors and game rules.

Detailed information, such as venue and schedule, will be announced by the POC of each country/region since the event format is different for each country/region.

(2) Qualification Round

- Teams will use the program that they developed for the preliminary round, although they may modify the program as needed.
- Teams will compete online using a simulator.
- Six finalists will be selected to go on to the ISS/Kibo competition.
- The scoring factors and game rules are the same as for the preliminary round.

Detailed information, such as venue and schedule, will be announced by Kibo-RPC secretariat.

Figure 1.6.3 Finalist Selection at Qualification Round

(3) Final Round

- Each team's program that was developed using JAXA's simulation environment will be uplinked to Astrobee.
- Teams need to complete their programs and submit them to Kibo-RPC secretariat.
- Teams may modify the program that they used in an earlier round.
- 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 2020

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 can't come to Japan.

Please ask the POC in each country/region about travel expenses and accommodation fee.

1.6.4. Release of Simulation Environment

The web-based simulation environment will be prepared in phases:

Please check the Kibo-RPC website.

- November 20th, 2019:   Initial release
- December 25th, 2019:   2nd update

1.7. Terms and Conditions

Applicants must agree with all of the terms below.

1.7.1. Privacy Policy

(1) Collected personal information will be used for the operation of the Kibo-RPC.

(2) The videos and photos taken at this tournament will be made publicly available.

1.7.2. Responsibility of Applicant and Exemption Clause of JAXA

(1) JAXA will not be liable for any problems that may occur during program creation, travel, etc. Applicants are fully responsible for solving issues of this nature that may arise.

(2) Applicants must guarantee that information on the application form and the program code do not violate any legal rights or obligations, such as intellectual property or export control, as applicable. If a legal problem related to the submitted information arises, the applicant will take full responsibility to solve the problem.

(3) If there is a violation of these matters registration may be canceled, even after registration.

1.7.3. Notes

(1) Underage applicants must first obtain parental consent.

(2) Do not use the simulator environment provided in this Kibo-RPC for any purpose other than the Kibo-RPC. Sign-on information for accessing the simulator environment will be made available only to the tournament applicants. Do not disclose this information to anyone else.

(3) JAXA may use the program codes and results of the competition for the purpose of public relations and education.

(4) All of the submitted program will be publicly opened prior to the final round of the competition.

2. Game Information

2.1. Scenario

A meteor shower entered the orbit of the International Space Station (ISS) 400 km above the ground. One of the meteors hit the ISS, and a part of an ISS module was damaged causing an air leak.

The damaged section turned out to be the Japanese experimental module Kibo. We need to identify the damaged area and repair it using Astrobee; the lower air pressure and debris are preventing astronauts from entering to make repairs. In order to identify the damaged place precisely, we must search in Kibo module and find the leak point by collecting information using Astrobee with the support of Int-Ball, the camera robot. Then we need to weld it closed with a laser. Without repair, the damage will spread. Time is limited!

When air leaks from a meteor puncture, the scattered debris scratches the walls and floors. Analyze each scratch and determine whether it was pre-existing or caused by the puncture. Until the hole is identified, we must continue our search so processing of the images from Astrobee's camera is indispensable. At the beginning of the exercise, the hole is small enough to be fixed with Astrobee's laser based on the rate at which the air pressure is dropping. However, if the Astrobee's laser targets the wrong spot, it may enlarge the hole.

Accuracy in finding the hole pointed to the scratches and positioning the laser pointer are the keys to stopping the air leak. Currently, astronauts are in the process of evacuation and flight controllers on the ground are busy providing support. We want to leave Astrobee's control to each team to save Kibo!

*This story is fiction.

2.2. Game Overview

Teams need to create a program that moves the Astrobee from the start position to the specified places, avoiding obstacles and pointing the laser at the target.

The score is calculated by the combination of the accuracy of laser pointing of Astrobee on the target and the elapsed time, and so on.

2.3. Competition Flow

2.3.1. Preliminary Round

(1) Game flow

Teams create a program to control NASA's Astrobee in JAXA's simulation environment.

1. Move Astrobee from the start position to P3 using QR codes at P1-1 to P2-3 in the absolute coordinate system.

2. Move Astrobee to P3 based on information obtained at P1-1 to P2-3, and you will be able to find the "Target Point" that represents the air leak point with Astrobee's camera. Approach Astrobee the target from P3 and stop it at the appropriate distance from the target point using the target's AR tag.

3. Use an image processing algorithm to illuminate the center of the target point with Astrobee's laser pointer. It will be necessary to allow for the offset between Astrobee's camera and the laser pointer.

   * The time limit is 10 minutes for one run. If the program does not complete the mission within 10 minutes, the action after 10 minutes will not be reflected in the score.

   Make sure your program complete the mission within the time limit.
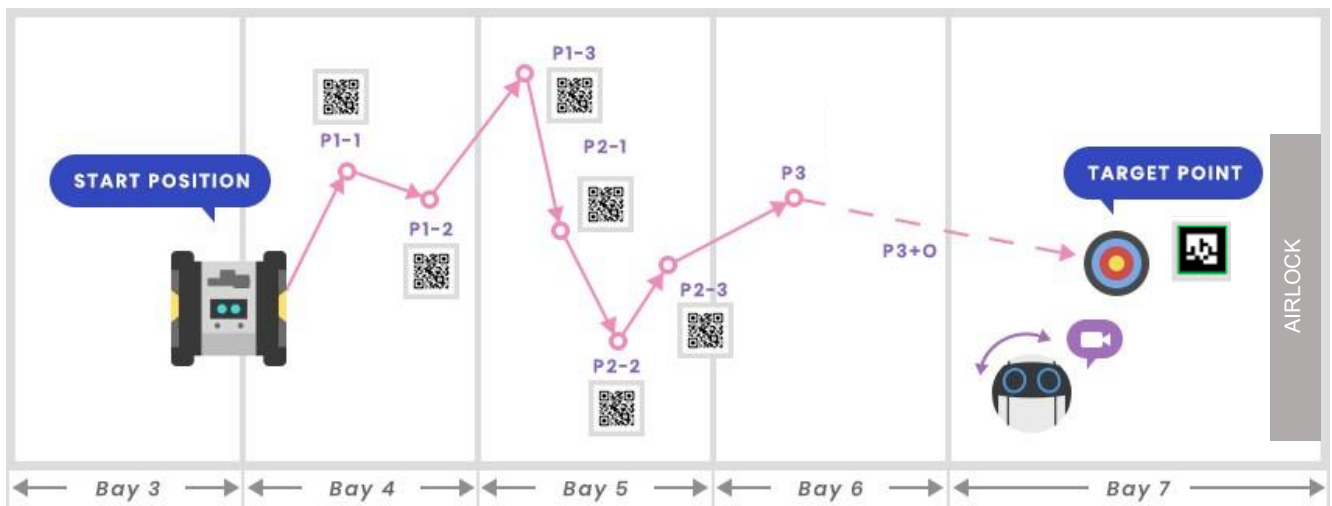
Figure 2.3.1-1 Game Outline of The Preliminary Round
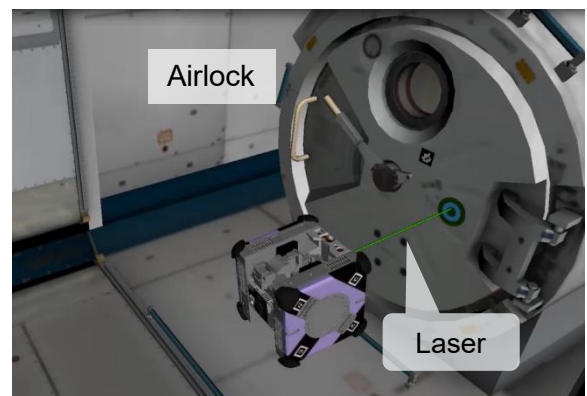
Figure 2.3.1-2 Astrobee Simulator Server Sample Screenshots

(2) Preconditions

Table 2.3.1-1 Preconditions of the Preliminary Round

#	Content
1	The starting position is fixed. The coordinates are; Position (x, y, z) = (10.95 -3.75 4.85) Orientation (x, y, z, w) = (0 0 0.707 -0.707) (Undock command is not needed.)
2	There is an AR tag near the target point. The position of the target point and an AR tag varies randomly.
3	P3 is a fixed position around the target point (the camera can capture the AR tag). QR codes at P1-1 to P2-3 provide information about P3.
4	About the positions of P1-1 to P1-3 and P2-1 to P2-3, please refer to (4) Object.
5	A Keep-Out Zone (KOZ)* that simulates an obstacle is set somewhere in the path from the start position to P3. This KOZ is given as a precondition. About the detail, please refer to (5) Keep-Out-Zone. * Astrobee cannot move into KOZ.
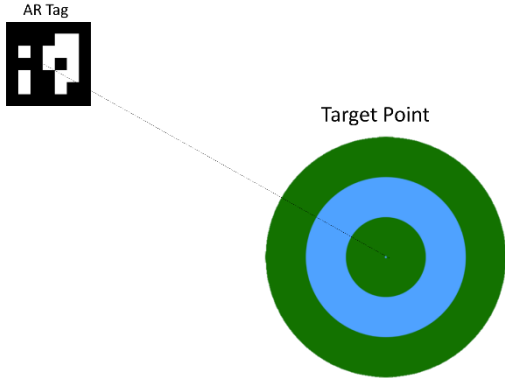
(3) Scoring Factors

Table 2.3.1-2 Scoring Factors of the Preliminary Round

| #	Factor	Detail
1	Arriving at transit point (P1-1 to P2-3)	You will gain points by reading QR code information correctly at each transit point.
2	Finding the target point	You will gain points by reading an AR tag ID correctly at each transit point.
3	Approaching the target point	You will gain points by aiming Astrobee's laser to the target point. The score may change depending on the accuracy of the laser aiming on the target point.
4	Elapsed time	You will be also judged by the time needed to complete the mission.

(4) Objects

Table 2.3.1-3 Objects of the Preliminary Round

#	Name of object	Method
1	QR code tag	The size of QR code tag is 20cm square. Information format is followings. Please create the program to move to P3 by quoting the values in [ ]. P1-1：pos_x, [P3 x coordinate]　(e.g. pos_x, 1.23) P1-2：pos_y, [P3 y coordinate]　(e.g. pos_y, 1.23) P1-3：pos_z, [P3 z coordinate]　(e.g. pos_z, 1.23) P2-1：qua_x, [P3 quaternion element x] (e.g. qua_x, 1.23) P2-2：qua_y, [P3 quaternion element y] (e.g. qua_y, 1.23) P2-3：qua_z, [P3 quaternion element z] (e.g. qua_z, 1.23) A dummy QR code ←20cm→ Table 2.3.1-4 Position Information of QR code tags <table><tr><td>No.</td><td>Position (x, y, z)</td><td>Orientation (x, y, z, w)</td></tr><tr><td>P1-1</td><td>(11.5, -5.7, 4.5)</td><td>(0, 0, 0, 1)</td></tr><tr><td>P1-2</td><td>(11, -6, 5.55)</td><td>(0, -0.7071068, 0, 0.7071068)</td></tr><tr><td>P1-3</td><td>(11, -5.5, 4.33)</td><td>(0, 0.7071068, 0, 0.7071068)</td></tr><tr><td>P2-1</td><td>(10.30, -7.5, 4.7)</td><td>(0, 0, 1, 0)</td></tr><tr><td>P2-2</td><td>(11.5, -8, 5)</td><td>(0, 0, 0, 1)</td></tr><tr><td>P2-3</td><td>(11, -7.7, 5.55)</td><td>(0, -0.7071068, 0, 0.7071068)</td></tr></table>
2	Target Point	The size of AR tag is 5cm square. -It is located 45° and 20cm upper left from target point. -"Aruco.DICT_5X5_250" is used as dictionary. The size of target point is 10cm circle. AR Tag　　　　Target Point Figure 2.3.1-3 Target point and AR code

(5) Keep-In-Zone (KIZ) and Keep-Out-Zone (KOZ)

Keep-In-Zone (KIZ) is defined as the area where Astrobee can move around, and basically set along the walls of Kibo. It is a preset boundary in Astrobee and If the destination of the moving path of Astrobee is outside the KIZ, that is rejected. You need to design each moving path of Astrobee within the KIZ.

The Keep-Out-Zone (KOZ) are set inside the KIZ as a volumetric zone and used as some obstacles inside Kibo in the Kibo-RPC. You need to design each moving path of Astrobee to avoid the KOZ. (Refer to Figure 2.3.1-4, 2.3.1-5, 2.3.1-6 and Table 2.3.1-5)
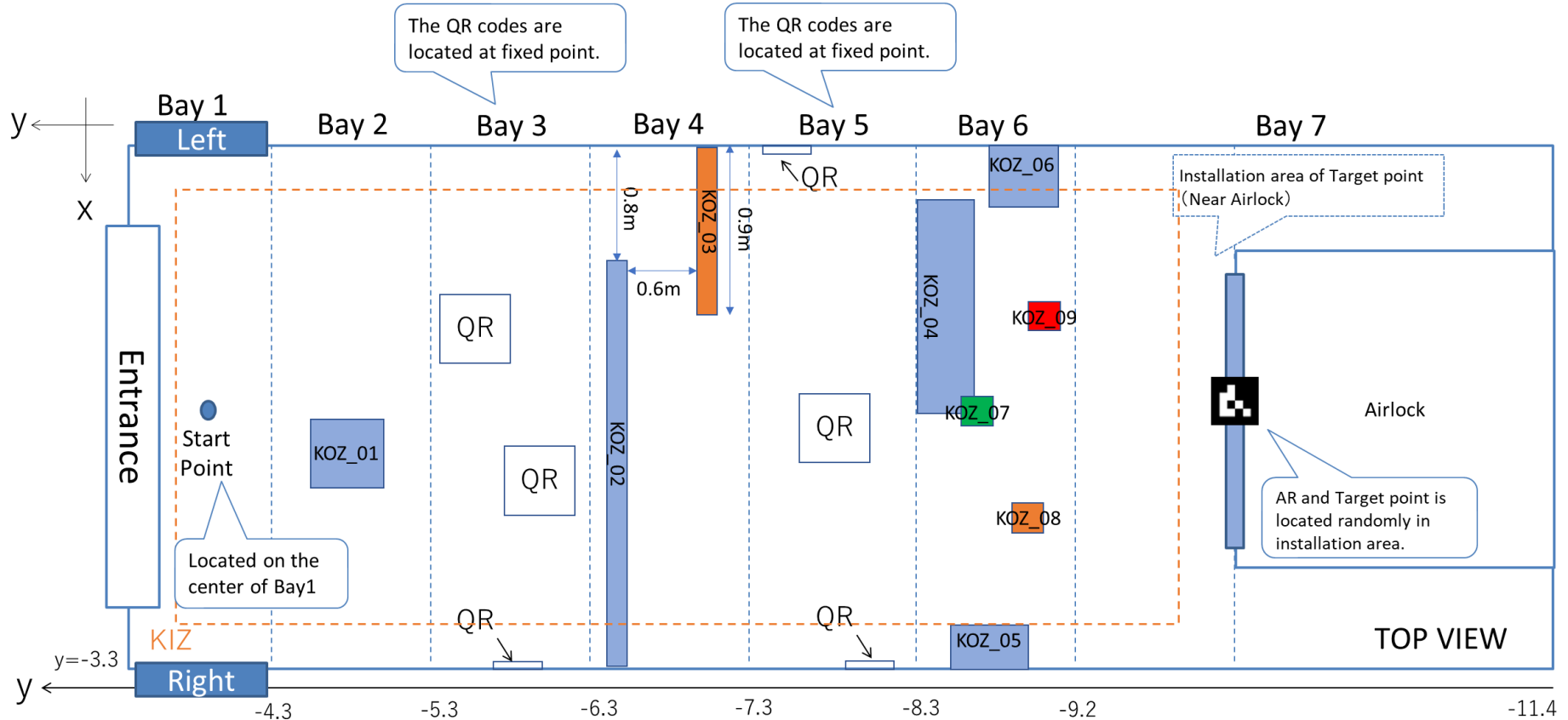
Top View of Kibo Module Inside

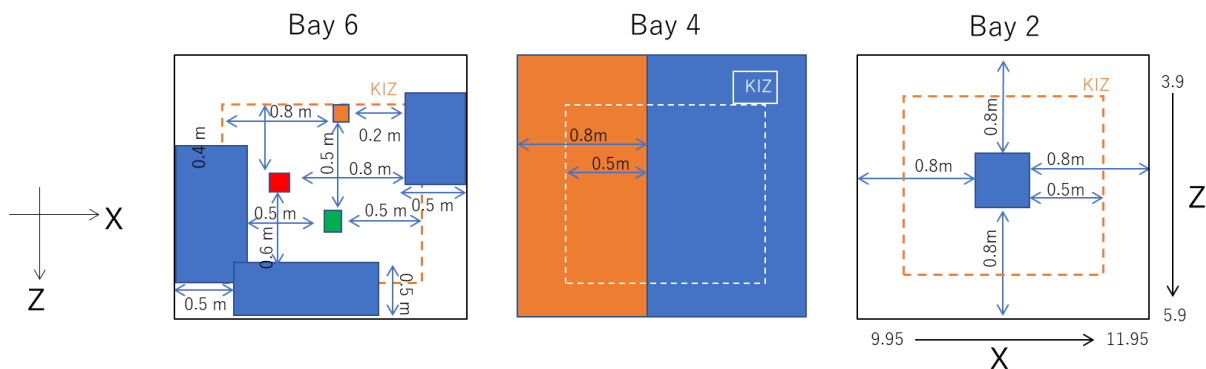Figure 2.3.1-4 KIZ and KOZ of The Preliminary Round (Top View)

Figure 2.3.1-5 KIZ and KOZ of The Preliminary Round (Sectional View)

Table 2.3.1-5 Installed Coordinate of Obstacles

#		x_min	y_min	z_min	x_max	y_max	z_max
	01	10.75	-4.9	4.8	10.95	-4.7	5.0
	02	10.75	-6.5	3.9	11.95	-6.4	5.9
	03	9.95	-7.2	3.9	10.85	-7.1	5.9
	04	10.10	-8.6	5.4	11.1	-8.3	5.9
KOZ_	05	11.45	-9.0	4.1	11.95	-8.5	5.1
	06	9.95	-9.1	4.6	10.45	-8.6	5.6
	07	10.95	-8.4	4.9	11.15	-8.2	5.1
	08	11.05	-8.9	4.2	11.25	-8.7	4.4
	09	10.45	-9.1	4.6	10.65	-8.9	4.8
KIZ		10.25	-9.75	4.2	11.65	-3	5.6

*The origin of the coordinate axis is set outside of Kibo

$$x_{min}, y_{min}, z_{min}$$
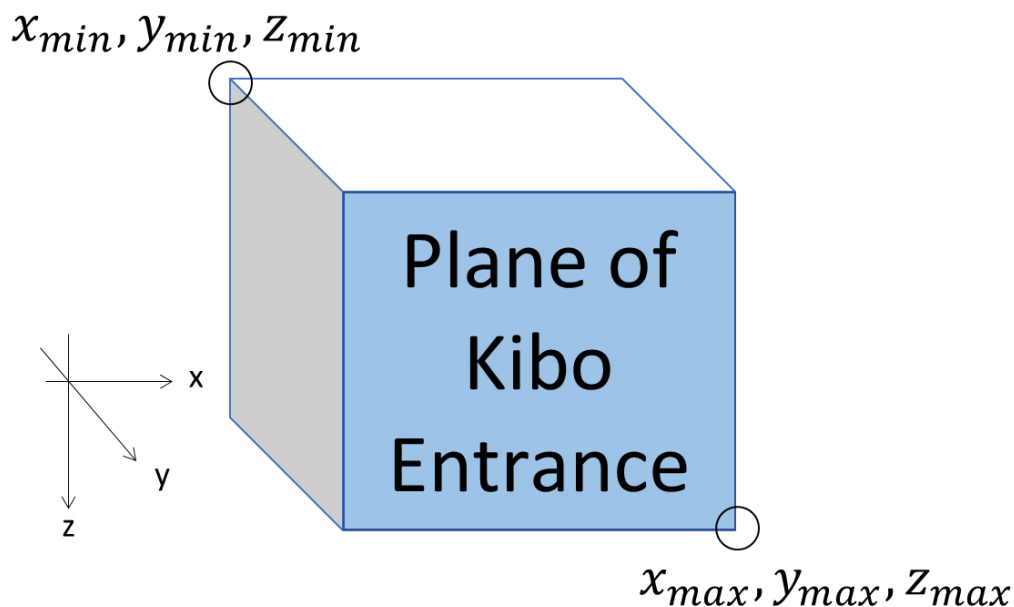

$$x_{max}, y_{max}, z_{max}$$

Figure 2.3.1-6 Definition of the coordinates

15

2.3.2. Qualification Round

This is the same as the preliminary round except that it is an international competition.

2.3.3. Final Round (Onboard)

(1) Game Flow

In the final game, teams will create a program that moves Astrobee onboard from the starting position to the target point by avoiding obstacles and point the laser at the target point.

The score is calculated from the accuracy of laser pointing at the target point, the elapsed time, and so on.

1. Move Astrobee from the start position to P3 through P1 and P2 in the absolute coordinate system by reading QR codes, which contain information about P3, using Astrobee's camera.

2. Move Astrobee to P3 based on the information obtained at P1 and P2, and you will be able to find the "Target Point" being the air leak with Astrobee's camera. Approach the target from P3 and stop Astrobee at the appropriate distance from the target point using the target's AR tag.

3. Use an image processing algorithm to illuminate the center of the target point with a laser pointer of Astrobee. It is needed to account for the offset between Astrobee's camera and the laser pointer.

   * The time limit is 10 minutes for one run. If the program does not complete the mission within 10 minutes, the action after 10 minutes will not be reflected in the score.

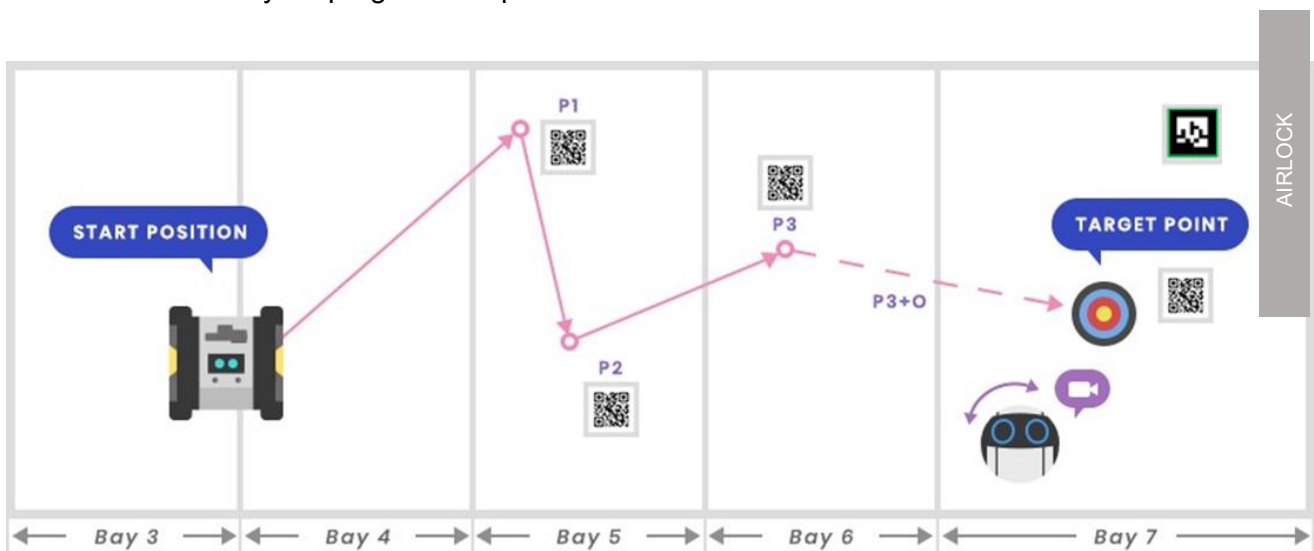   Make sure your program complete the mission within the time limit.

Figure 2.3.3-1 Game Outline of The Final Round

(2) Preconditions

Table 2.3.3-1 Preconditions of the Final Round

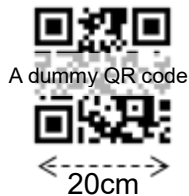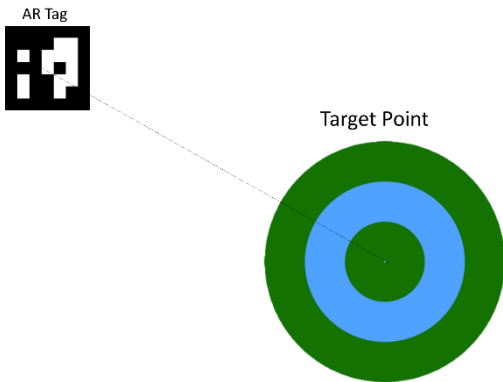| #	Content
1	The starting position is fixed and announced to participants. (Undock command is not needed.)
2	There is an AR tag near the target point. The position of target point and an AR tag is fixed and not announced to participants.
3	P3 is a fixed position near the target point (the camera can capture the AR tag). QR codes at P1 and P2 provide information about P3.
4	About the positions of P1 and P2 , please refer to (4) Object.
5	A Keep-Out Zone (KOZ)* that simulates obstacles are set somewhere in the path from the start position to P3. This KOZ is given as a precondition. About the detail, please refer to (5) Keep-Out-Zone. * Astrobee cannot move into KOZ.

(3) Scoring Factors

Table 2.3.3-2 Scoring Factors of the Final Round

#	Factor	Detail
1	Arriving at transit points (P1, P2)	You will gain points by reading QR code information correctly at each transit point.
2	Finding the target point	You will gain points by reading an AR tag ID correctly at each transit point.
3	Approaching to the target point	You will gain points by aiming Astrobee's laser at the target point. Score may change depending on the accuracy of the laser pointing at the target point.
4	Elapsed time	You will be also judged by the time needed to complete the mission.

(4) Objects

Table 2.3.3-3 Objects of the Final Round

#	Name of object	Method
1	QR code tag	The size of AR tag is 20cm square. Information format is followings. P1：pos_x, [P3 x coordinate], pos_y, [P3 y coordinate], 　　　pos_z, [P3 z coordinate] 　　　(e.g. pos_x, 1.23, pos_y, 2.34, pos_z, 3.45) P2：qua_x, [P3 quaternion element x], 　　　qua_y, [P3 quaternion element y], 　　　qua_z, [P3 quaternion element z], 　　　(e.g. qua_x, 1.23, qua_y, 2.34, qua_z, 3.45) A dummy QR code　20cm Table 2.3.3-4 Position information | No.	Position (x, y, z)	Orientation (x, y, z, w)
---	---	---
P1	(11.5, -5.7, 4.5)	(0, 0, 0, 1)
P2	(10.30, -7.5, 4.7)	(0, 0, 1, 0)	
2	Target Point	The size of AR tag is 5cm square. It is located 45° and 20cm upper left from target point. "Aruco.DICT_5X5_250" is used as dictionary. The size of target point is 5cm circle. AR Tag Target Point Figure 2.3.3-2 Target point and AR code

(5) Keep-In-Zone (KIZ) and Keep-Out-Zone (KOZ)

TBD

2.4. Tips for Astrobee Characteristics

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

2.4.1. Rendering of Astrobee

Figure 2.4.1 indicated Astrobee 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.
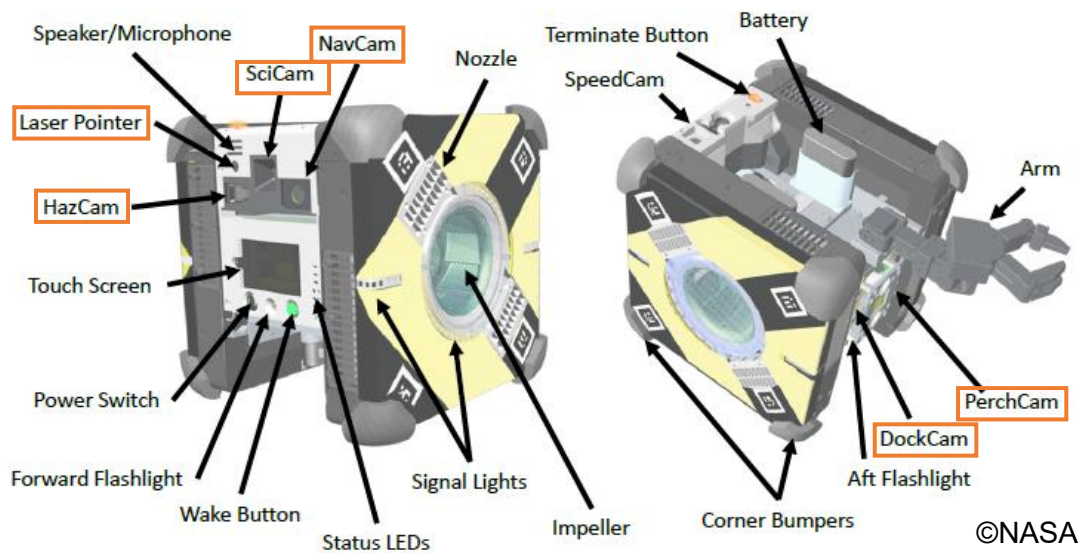

Figure 2.4.1-1 Rendering of Astrobee

Table 2.4.1-1 Equipment on Astrobee

| Name	Explanation
HazCam	A monochrome camera for detecting obstacles within 30cm
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. 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 dock 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 need.)
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)
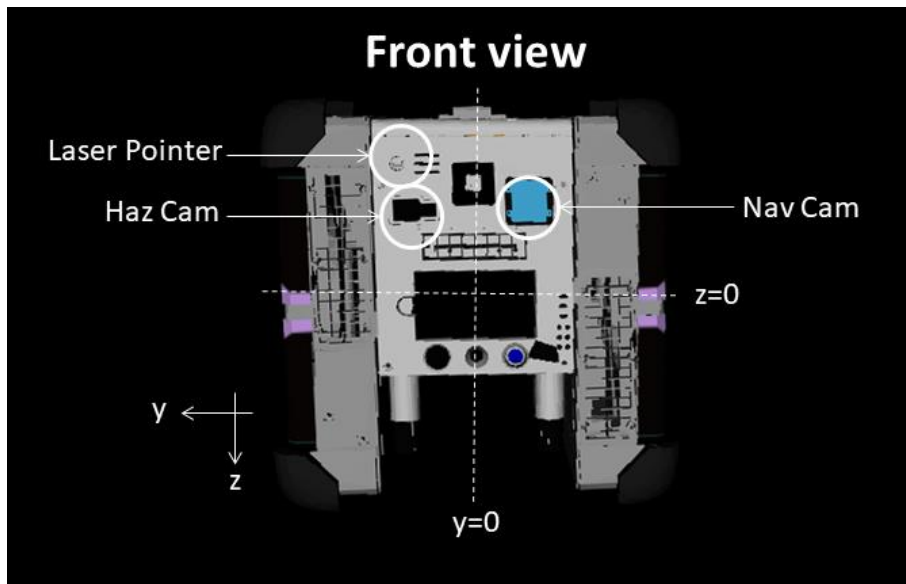
Figure 2.4.1-2 Astrobee Front View

Table 2.4.1-2 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
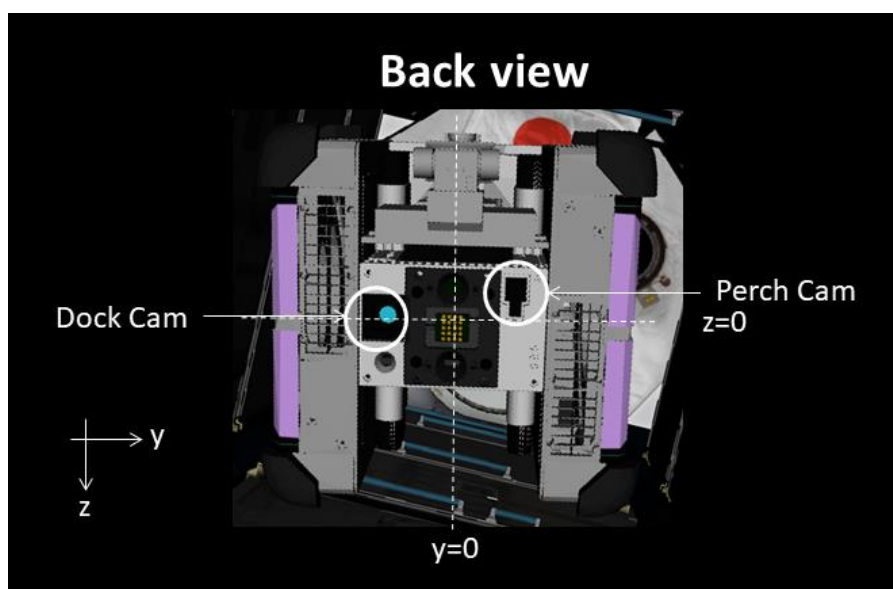
Figure 2.4.1-3 Astrobee Back View

Table 2.4.1-3 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

2.4.2. 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 detect 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.