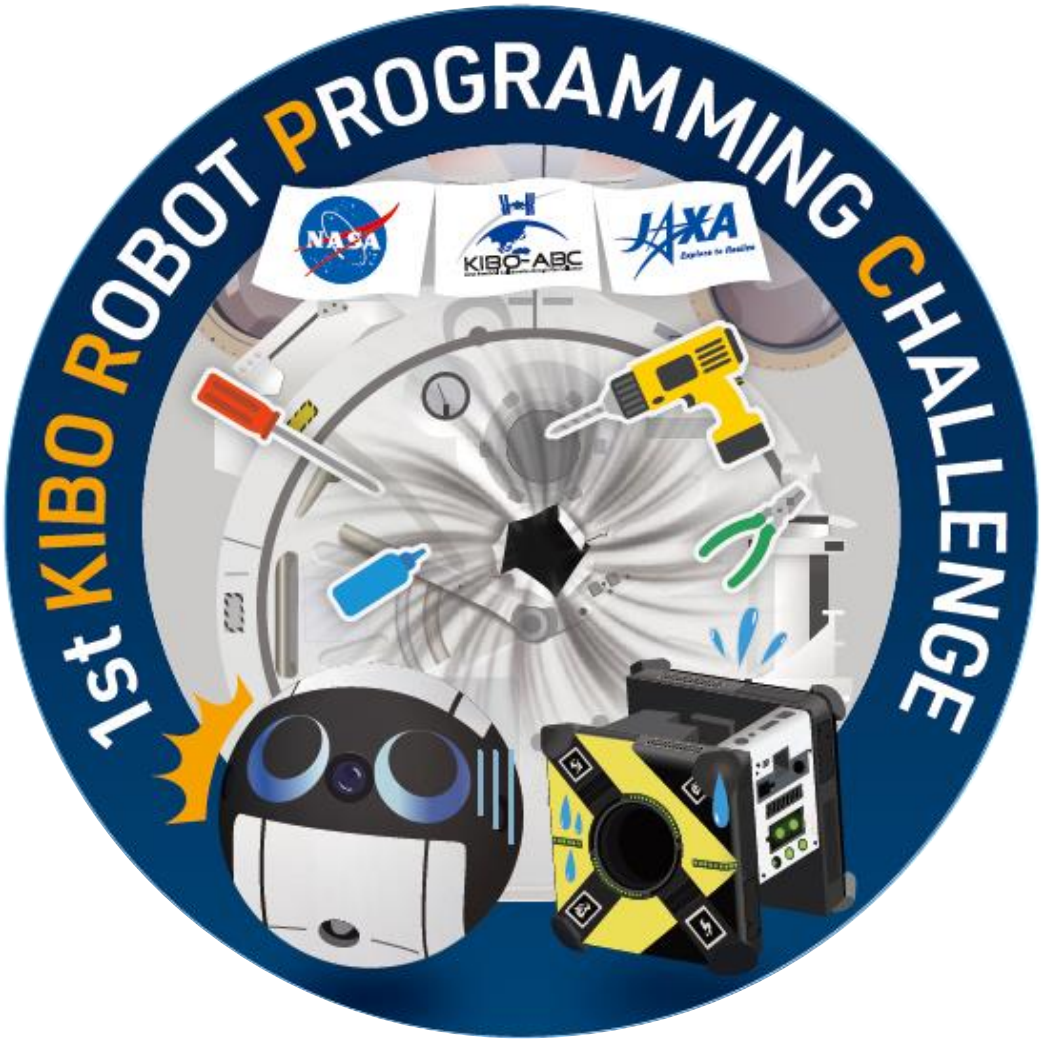
Version 1.5
Released Date: September 30th, 2020

1st Kibo Robot Programming Challenge

Final Round Rulebook



Version 1.5 (Released Date: September 30th, 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
June 25th, 2020	1.0	All	-
June 29th, 2020	1.1	3.6	Modify the coordinates of P1 and P2.
July 15th, 2020	1.2	All	Change of document style and expression
		2.1	Change of program submission deadlines
		2.3	Addition of program submission method
		3.6	Error correction regarding P2 orientation (No changes in simulator)
		3.7	Modify KIZ/KOZ (simulator is also updated)
		5.1(4)	Addition of warning regarding programming.
August 5th, 2020	1.3	5.1(3)	Addition of setting of app_name
August 21st 2020	1.4	2.1	Extension of APK submission deadline
		3.6	Error correction regarding P1 orientation (No changes in simulator)
		5.1(3)	Error correction regarding application ID (“jp” was missing)
September 30th 2020	1.5	2.1	Change of program submission deadlines
		3.6、3.7	Modify the coordinates of P1 and P2 and KIZ/KOZ

Contents

1. Introduction

In the Final Round, national teams compete each other to win for the world championship by using free-flyer robots, Int-Ball and Astrobee, in the ISS/ Kibo module!

The winning teams of the Preliminary Round in each country/region are eligible to participate in the Final Round. The contestants develop the program (Android Application Package: APK) using JAXA's simulation environment to run Astrobee on-orbit before the Final Round. Please refer to the Programming Manual for how to develop the application. This Guidebook shows rules and details of the Final Round.

2. Joining Final Round

2.1. Participation of the Final Round

Only the winner of the Preliminary Round in each country/region can participate in the Final Round. In the Final Round, the participants need to refine the configuration from the Preliminary Round. Please create a program for the finals and submit APK and source code by the submission deadline. Refer to "Submission of Final Round APK" in section 2.3.

On-orbit Rehearsal is carried out approximately one month before Final Round, and programs of several teams are picked up and run. The programs will be selected randomly, and no specific feedback will be given to participants unless there are significant issues to all participants.

1) Draft Program Submission Deadline:
23:59 July 26, 2020 (JST)

2) Final Round Program Submission Deadline:
10:00AM October 5, 2020 (JST)

2.2. Refining the Program

Participants cannot use the same APK as the Preliminary Round because the Final Round configuration is different from the Preliminary Round. Find the configuration settings for the Final Round in Section 3.

For program refinement, participants can use a web simulator on the "Final Refurbish" tab. Also, please read and adhere to Section 5.1 carefully; this is the critical note for safe and smooth operation in the ISS.

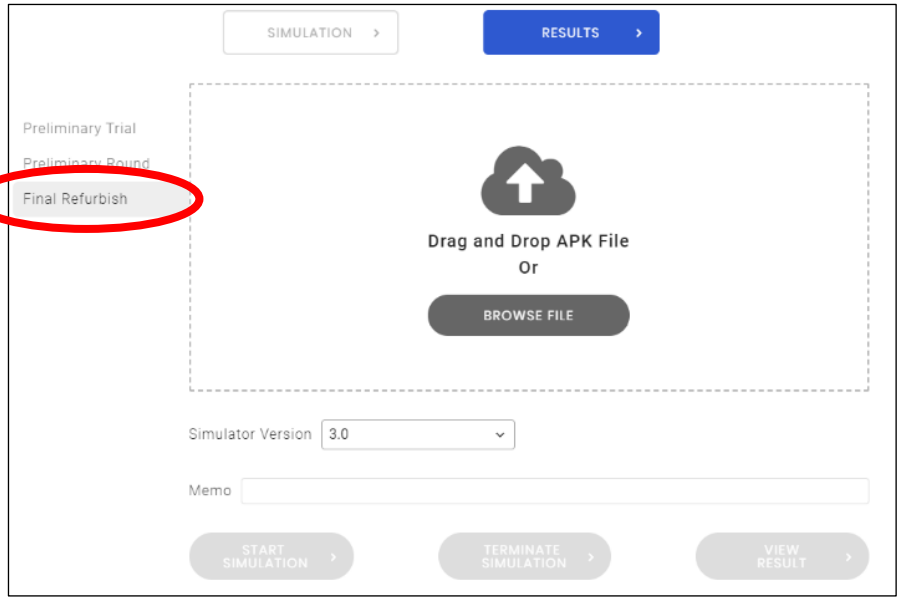


Figure 2.2-1 Simulator UI for Final Refurbish

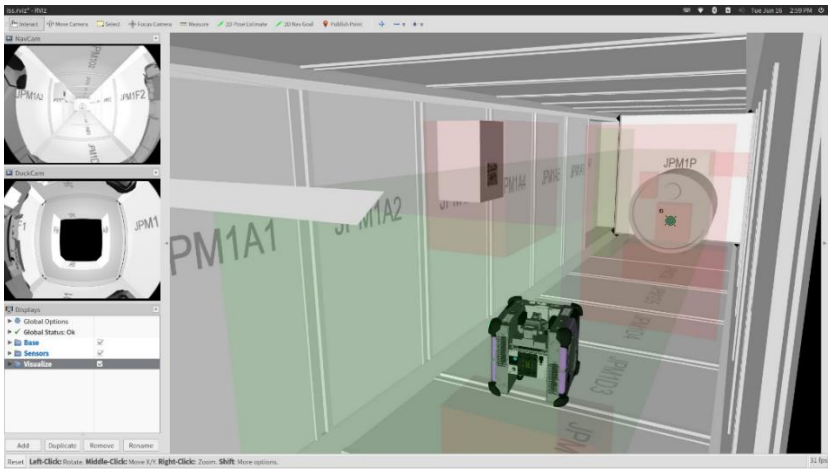


Figure 2.2-2 Simulator to refurbish APK for Final Round

2.3. Submission of Final Round APK and Source Code

The contestants need to create and submit a program for Final Round in advance by the deadline. In addition, the source code shall be confirmed with JAXA and NASA in advance for safety reasons since we use the Astrobee on-orbit for the Final Round. Therefore, you must submit (1) APK and (2) source code.

Note: Camera parameters and finish API are different between simulator and on-orbit. Please refer to 5.1(5), and do not forget to switch these settings before submission.

(1). How to Submit APK

You can select one APK and press submit button on “RESULTS” display. Only class E or better, i.e., those that are guaranteed to have been created in accordance with appropriate procedures, can be submitted. (If you have properly adjusted the parameters and API for Astrobee actual machine as section 5.1(5), the simulation result may not be class A.)

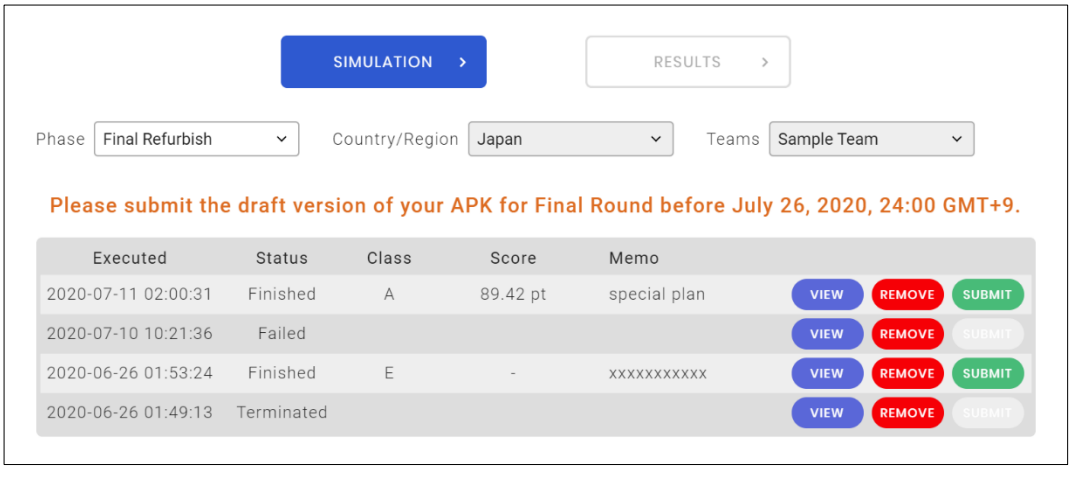


Fig.2.3-1 How to Submit APK

(2). How to Submit Source Code

You must submit your source code via e-mail. Follow the instructions below.

1. Generate MD5 of APK file

Kibo-RPC secretariat confirms APK submitted via website and this md5.

A) on Windows

Execute following commands in command prompt.

> cd [path to apk directory]

> certutil -hashfile [apk file name] MD5 > apk.md5

“apk.md5” is created and it includes 32-digit hash value.

(e.g.)

> cd C:\DefaultApk\app\build\outputs\apk\

> certutil -hashfile app-debug.apk MD5 > apk.md5

B) on Ubuntu

Execute following commands in terminal.

$ cd [path to apk directory]

$ md5sum [apk file name] > apk.md5

“apk.md5” is created and it includes 32-digit hash value.

(e.g.)

$ cd ~/DefaultApk/app/build/outputs/apk/

$ md5sum app-debug.apk > apk.md5

2. Delete APK and large file/directories

Delete following file. (Be sure not to delete md5)

- [root dir]/app/build/outputs/apk/app-debug.apk

Delete following directories.

- [root dir]/app/build/generated/
- [root dir]/app/build/intermediates/
- [root dir]/app/build/tmp/
- [root dir]/.gradle/

3. Compress (zip, tar, etc.) root directory and send the compressed file to secretariat

Compressed file should be a few hundred KB or a few MB. Confirm that all files, especially Java source files and md5, are included in the compressed file.

Please send it to Z-KRPC@ml.jaxa.jp

3. Settings and Conditions of ISS

3.1. Game Flow

In the Final Round, each team creates a program that moves Astrobee onboard from the starting position to the target point by avoiding KOZ and illuminate the target point with the laser.

The calculation of the score is by using the accuracy of laser pointing and the elapsed time.

1. Move Astrobee from the start position to P3 through P1 and P2 in the absolute coordinate by reading QR codes, which contain information about P3.
2. Move Astrobee to P3 which obtained information at P1 and P2, and find the target point leaking air. Approach the target from P3 and stop Astrobee at the appropriate distance from the target point by using the target's AR tag.
3. Illuminate the center of the target point by using an image processing algorithm with a laser pointer of Astrobee. (You need to offset the distance between Astrobee's camera and the laser pointer.)

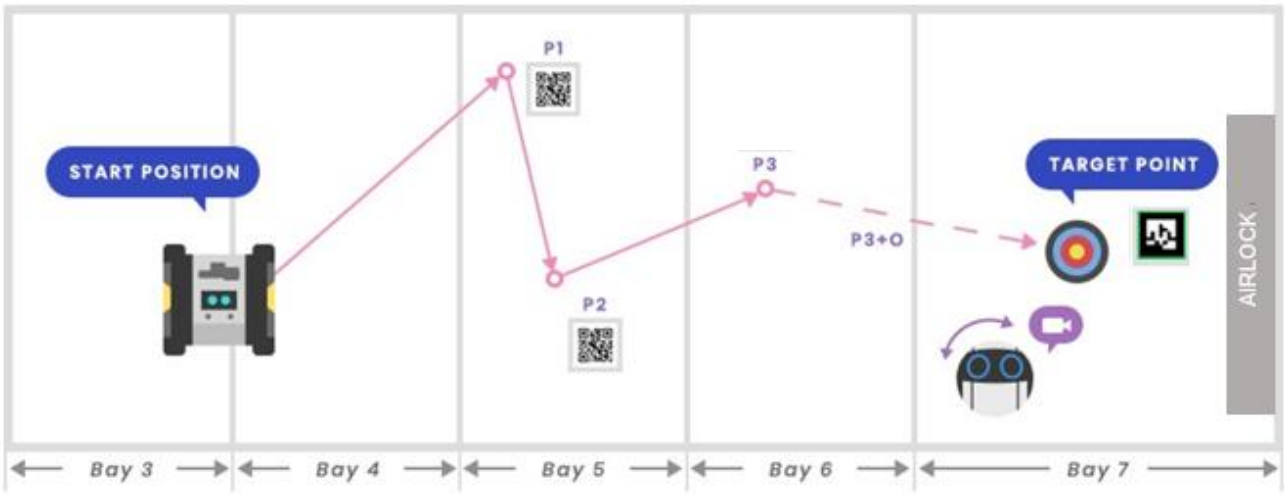


Figure 3.1-1 Game Outline of the Final Round

3.2. Preconditions

Table 3.2 Preconditions of the Final Round

#	Content
1	The starting position is fixed. The coordinates are; Position (x, y, z) = (10.95 -4.5 4.95) 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 positions of target point and an AR tag are fixed and not announced to participants.

3	P3 is a fixed position near the target point (the camera can read the AR tag). QR codes at P1 and P2 provide information about P3.
4	About the positions of P1 and P2, please refer to 4.5 Object.
5	A Keep-Out Zone (KOZ)* that virtually obstacles are set somewhere in the path from the start position to P3. This KOZ set as a precondition. Please refer to 3.7 Keep-Out-Zone for details. * Astrobee cannot move into KOZ.

3.3. Rules for the Final Round

Three standard rules for the Final Round follow.

(1) Only one APK, only one run

Each team submits one APK like the preliminary round. However, it only runs one-time in ISS. Since the Final Round uses Astrobee on orbit, it cannot be redone or interrupted. It is a one-time shot.

(2) The Time limit 10 minutes per team

If it exceeds 10 minutes, APK will automatically shut down. Please make sure that it completes the mission within 10 minutes. If Astrobee gets stuck or its self-position is lost, it automatically judges as a game-over.

(3) Operation of the APK on the day of the Final Round

Participants may NOT operate the APK on the day of the Final Round.

The submitted APKs are code-reviewed by the technical team of JAXA / NASA and uplinked to Astrobee on orbit in advance. One team can run only once. The APKs are started with the execution command sent from ground operators.

(4) Judging method

On the ISS orbit, the speed and accuracy of the mission achievements are judged by the following methods. Details of the scoring criteria are given in Sections 3.4 and 3.5.

Speed: Time from APK execution start to send Finish command is recorded as the time stamp in Astrobee, which is available to the ground as telemetry.

Accuracy: Photo and/or video at the laser irradiation is recorded in ISS.
The accuracy of this photograph is judged by image analysis.

Besides, the level of mission achievement is judged in the same way as the Preliminary Round, using APIs (judgeSendStart, judgeSendDiscoveredQR, judgeSendDiscoveredAR, and judgeSendFinishISS).

3.4. Scoring Factors

#1-2 on the following table refers classification. #3-4 effect once a team reaches Class A.

Table 3.4-1 Scoring Factors of the Final Round

#	Factor	Detail
1	Arriving P1 and P2	Points are given by reading information from QR code correctly at each point.
2	Finding the target point	Points are given by reading an AR tag correctly.
3	Approaching the target point	Points by illuminating the target point with Astrobee's laser. Score change depending on the accuracy of the laser pointing.
4	Elapsed time	Points are given by the time using for completing the mission.

3.5. Ranking Method

The results are categorized by the level of mission achievement (See Table 3.5-1) with the run scored, and a team which marked the higher score in the same class become high rank.

Rank is given in the order of class A > B > C > D > E (See Table 3.5-2).

Table 3.5-1 Classification by achievement

Class	A	B	C	D	E
Finding P1	○	○	○	△ (1 QR found)	×
Finding P2	○	○	○		
Move to P3	○	○	×	-	-
Laser irradiation	○	×	-	-	-

Class A：Complete all missions. Found all the QR codes for P1 and P2, moved to P3 and finding AR-tag, then illuminated the target with the laser within 10 minutes
It is ranked in the order of the points obtained by the scoring criteria. Score range is from 0 to 100 points and calculated from the accuracy of laser pointing and elapsed time. The laser accuracy and elapsed time are weighted in these criteria.

Class B：Found AR-tag at P3, but laser illumination failed. It is ranked in the order of the time until moving to P3.

Class C：Found all the QR codes for P1 and P2.

It is ranked in the order of time until finding P1 and P2.

Class D：Found the QR code for P1 or P2.

It is ranked in the order of time until finding the QR code.

Class E：All missions failed. The team could not find a QR code. No score.

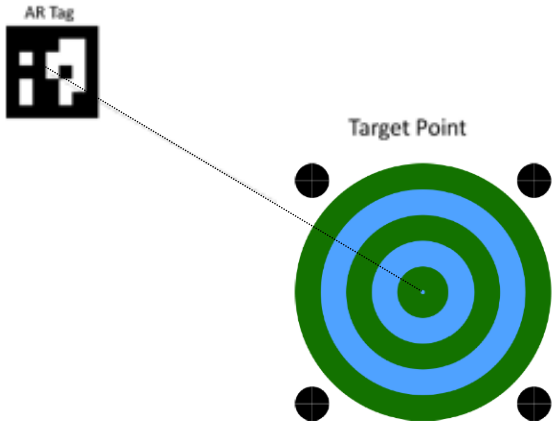
Table 3.5-2 Ranking and Class relation

Ranking	Team	Class	Score
1st	Team Q	A	80 pt
2th	Team R	A	32 pt
3th	Team T	B	6:14
4th	Team U	C	9:15
5th	Team H	D	1QRs 8:05
6th	Team X	D	1QRs 9:07
7th	Team P	E	-

Since the environmental conditions are different from in simulation and on-orbit, it is important to create a program that can show high performance even in the real environment.

3.6. Objects

Table 3.6-1 Objects of the Final Round

#	Name of object	Method
1	QR code tag	The size of QR code is 20cm square in the simulator but 7cm on orbit. Either QR codes can be read from P1 and P2. 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)  You can find QR codes at the position and orientation below. Table 3.6-2 Position information No. / Position (x, y, z) / Orientation (x, y, z, w) P1 / (10.7, -5.57, 4.5) / (0, 0, 1, 0) P2 / (11.1, -7.98, 5.4) / (-0.5, -0.5, -0.5, 0.5) * There is a possibility of adjustments after on-orbit verification.
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 7.5cm radius.  Figure 3.6-1 Target point and AR code

3.7. Keep-In-Zone (KIZ) and Keep-Out-Zone (KOZ)

Keep-In-Zone (KIZ) defines as the area where Astrobee can move around and 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. It would be helpful If the team designed 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 3.7-1, 3.7-2, 3.7-3 and Table 3.7-1)

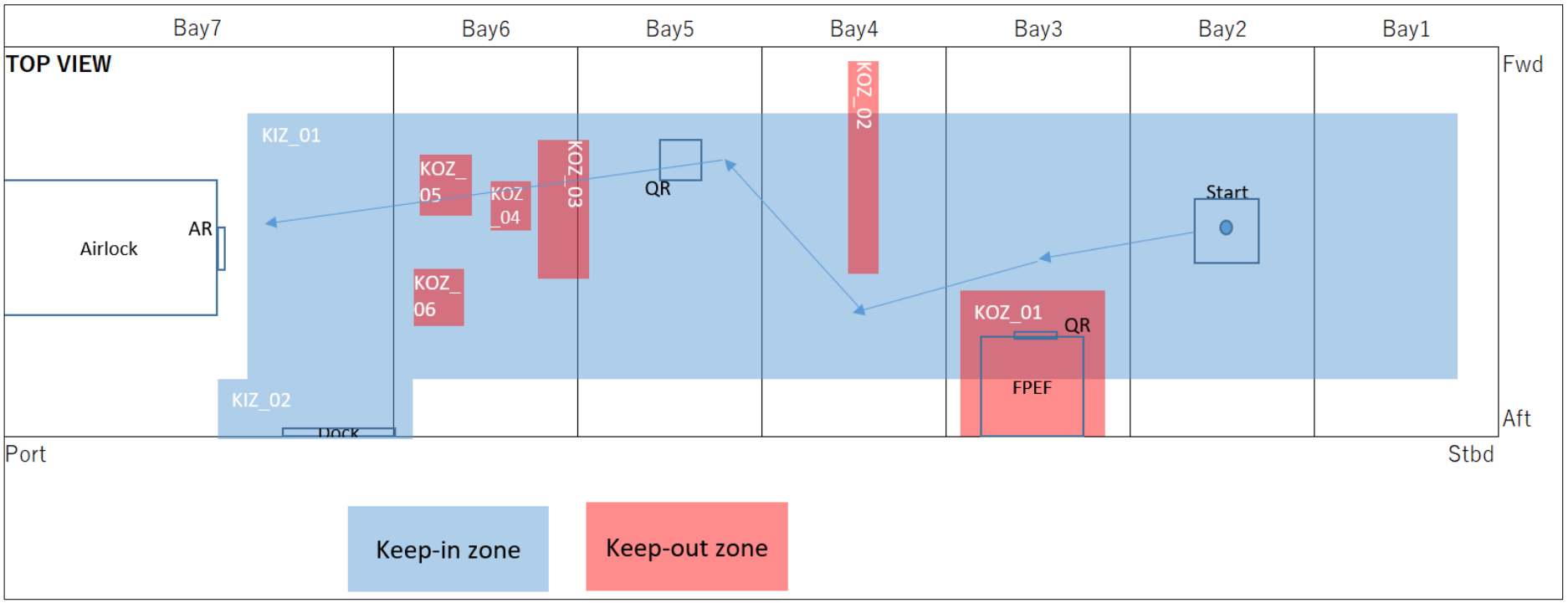


Figure 3.7-1 KIZ and KOZ of the Final Round (Top View)

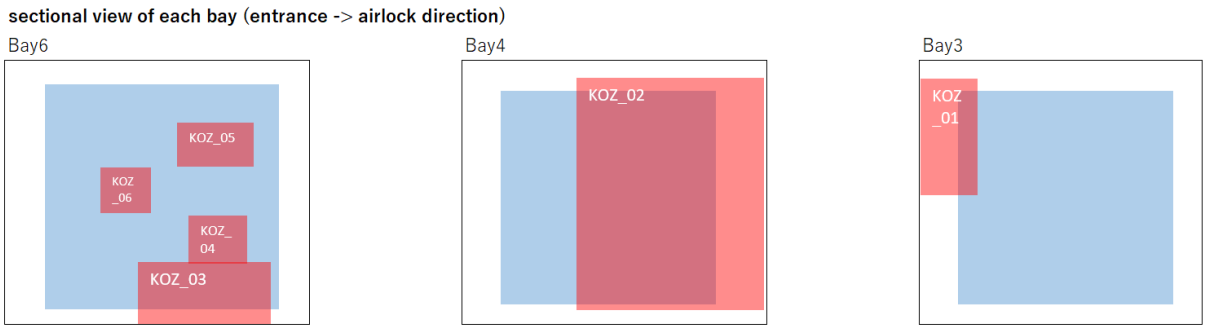


Figure 3.7-2 KIZ and KOZ of the Final Round (Sectional View)

Table 3.7-1 Installed Coordinate

#		x_min	y_min	z_min	x_max	y_max	z_max
KOZ_	01	10	-5.9	4.02	10.6	-5.09	4.88
	02	10.61	-6.58	4.02	11.85	-6.48	5.87
	03	10.68	-8.5	5.4075	11.85	-8.18	5.87
	04	11	-8.75	5.1	11.4	-8.5	5.4
	05	11	-9	4.32	11.55	-8.75	4.6
	06	10.68	-9.25	4.6	10.9	-9	4.945
KIZ	01	10.3	-10.2	4.32	11.55	-3.1	5.57
	02	9.5	-10.5	4.02	10.3	-9.6	4.8

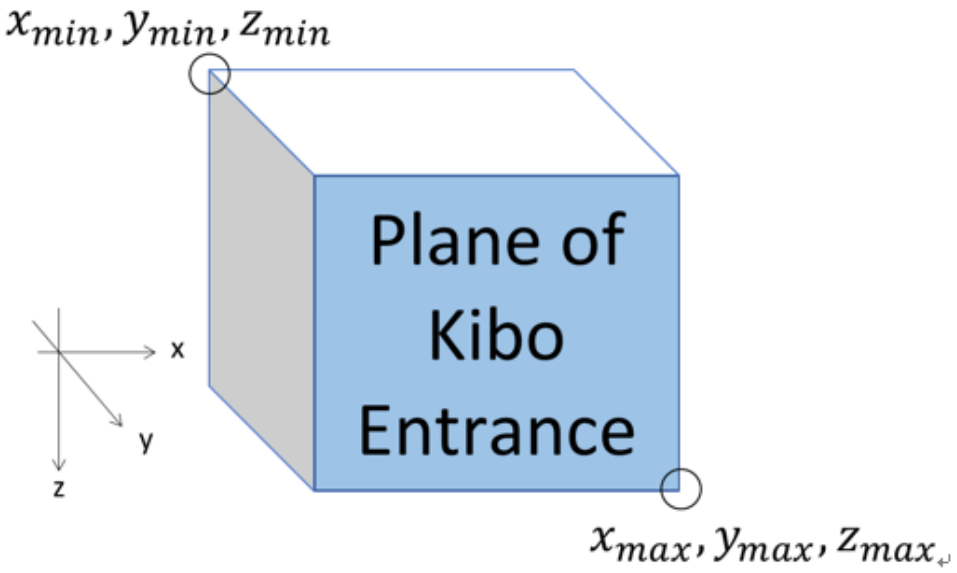


Figure 3.7-3 Definition of the coordinates

4. Event Content

This section describes the Final Round events held in real-time in JAXA.

4.1. Event Summary

The Final Round takes place at Tsukuba Space Center, and Astrobee is operated remotely by connecting Tsukuba Space Center and the ISS.

Tsukuba Space Center: https://global.jaxa.jp/about/centers/tksc/

Participants submit the APK on the Web by the deadline (Section 2.1), then the APK will be code reviewed and uplinked to install in the Astrobee in advance. On the day of the event, participants may come to the Tsukuba Space Center to watch the real-time event. If it is difficult to come to the venue, each team's leader should let the Kibo-RPC secretariat know to coordinate an alternative way, such as using a meeting tool. Figure 4.1-1 shows the general flow.

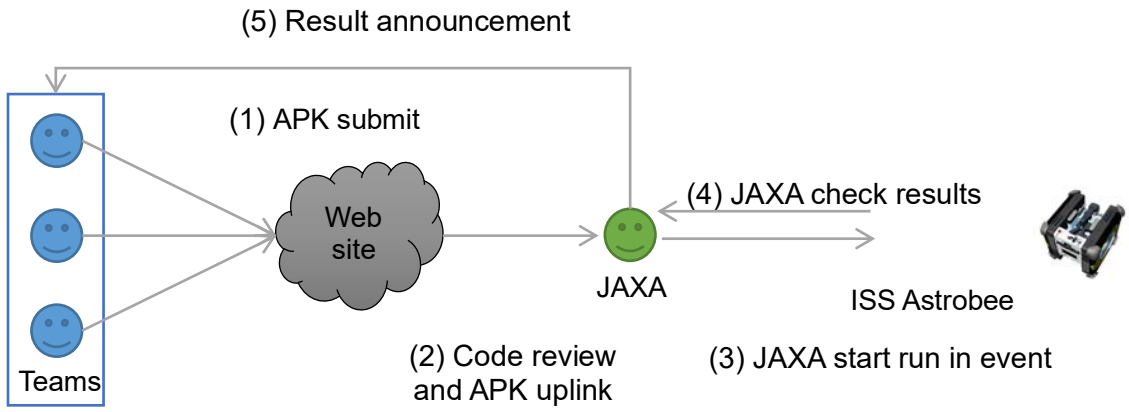


Figure 4.1-1 Real Event flow

4.2. Agenda

We will post the timetable on the website as soon as the content of the event is finalized.

5. Notes

5.1. About Final Round APK

This section describes the special notes for the Final Round APK.

(1) Infinite loop prohibited

Be sure to follow Section 5.1 of the Programming Manual. The infinite loop could get Astrobee stuck, and it might affect the final round event operation.

(2) Attention to computing resources

If the computing loads are high, Astrobee might be overloaded and not work on orbit. The specifications of Astrobee's HLP are as follows. Note that available resources are different by other software working on HLP. Multithreading is not recommended.

CPU: Qualcomm Snapdragon 820 (4 cores, 2.2GHz)

RAM: 4GB

(3) Setting the application ID

Each Final Round APK must have a unique application ID to avoid conflict when installing on Astrobee in the ISS. Follow the instructions in Section 3.2.3 of the Programming Manual and set your country name as APK name as follows:

[Australia] jp.jaxa.iss.kibo.rpc.australia

[Indonesia] jp.jaxa.iss.kibo.rpc.indonesia

[Japan] jp.jaxa.iss.kibo.rpc.japan

[Singapore] jp.jaxa.iss.kibo.rpc.singapore

[Taiwan] jp.jaxa.iss.kibo.rpc.taiwan

[Thailand] jp.jaxa.iss.kibo.rpc.thailand

[United Arab Emirates] jp.jaxa.iss.kibo.rpc.uae

Also please follow the Step.10 in Section 3.2.3 of the Programming Manual and set app_name as follows.

[Australia] australia

[Indonesia] indonesia

[Japan] japan

[Singapore] singapore

[Taiwan] taiwan

[Thailand] thailand

[United Arab Emirates] uae

If you have made the setting changes correctly, your APK should work on the web simulator.

(4) Cautions when irradiating laser

Please use the laser only to illuminate the Target. Due to safety reasons, it is prohibited to point the laser at the crew. It is required to inform the intended timing and target of laser irradiation in advance to both flight controllers and the crew. Therefore, create a program where the laser does not hit the crew and does not illuminate a place unnecessarily. DO NOT irradiate the laser blindly, for example, when reading QR/AR is failed. Please check your code not to proceed the processing despite the direction to aim is uncertain.

(5) Camera parameter and API change in an actual machine

There ae some differences in the orbital Astrobee actual machine in the camera parameters and Finish API used in the Simulator. Those parameters need to adjust before the APK submission. It is highly recommended to code with constants and flags for easy switching.

Table 5.1-1 Parameter

		Simulator	On-orbit
camera parameter	camera matrix	[344.173397, 0.000000, 630.793795, 0.000000, 344.277922, 487.033834, 0.000000, 0.000000, 1.000000]	NavCam: [692.827528, 0.000000, 571.399891, 0.000000, 691.919547, 504.956891, 0.000000, 0.000000, 1.000000] DockCam: [936.510555, 0.000000, 553.417823, 0.000000, 927.850939, 554.728586, 0.000000, 0.000000, 1.000000]
	distortion coefficients	[-0.152963, 0.017530, -0.001107, -0.000210, 0.000000]	NavCam: [-0.312191, 0.073843, -0.000918, 0.001890, 0.000000] DockCam: [-0.411405, 0.177240, -0.017145, 0.006421, 0.000000]
Finish API		judgeSendFinishSimulation()	judgeSendFinishISS()

(6) Performance of Localization

There is a possibility of losing Astrobee's self-position on-orbit. Once the self-position is lost, Astrobee may not be able to recover on its own. In this case, it means it is a game over. Be aware that this incident is not in the simulator.

It is well known that the technology Astrobee uses is likely to lose its self-position when the camera view gets too close to wall, floor, airlock, and so on because the camera cannot capture enough features at those places. Please note the above when creating your program.

(7) Code review

Before the submitted APK is uplinked to Astrobee on orbit, JAXA / ARC performs a code review for safety confirmation in advance. In the code review, if there is an inappropriate code in the submitted APK, we might delete it or instruct the participants to rewrite it.

5.2. On-orbital Events

This section describes the restrictions and special notes for on-orbit events.

(1) LOS occurrence

Communication with the orbit is via the United States ground station and the United States data relay satellite (TDRS). Therefore, it cannot sometimes communicate with the ground, called LOS (Loss of Signal). We try not to be in LOS while the participant's programs are running, but there are cases such as sudden LOS and instances where it cannot avoid. In that case, you may not be able to see a part of your Run.

(2) Emergency response

In real-time operations, the Final Round event may be suddenly cancelled or postponed due to emergencies on ISS. (i.e., the failure of the ISS and/or JEM critical equipment has occurred)