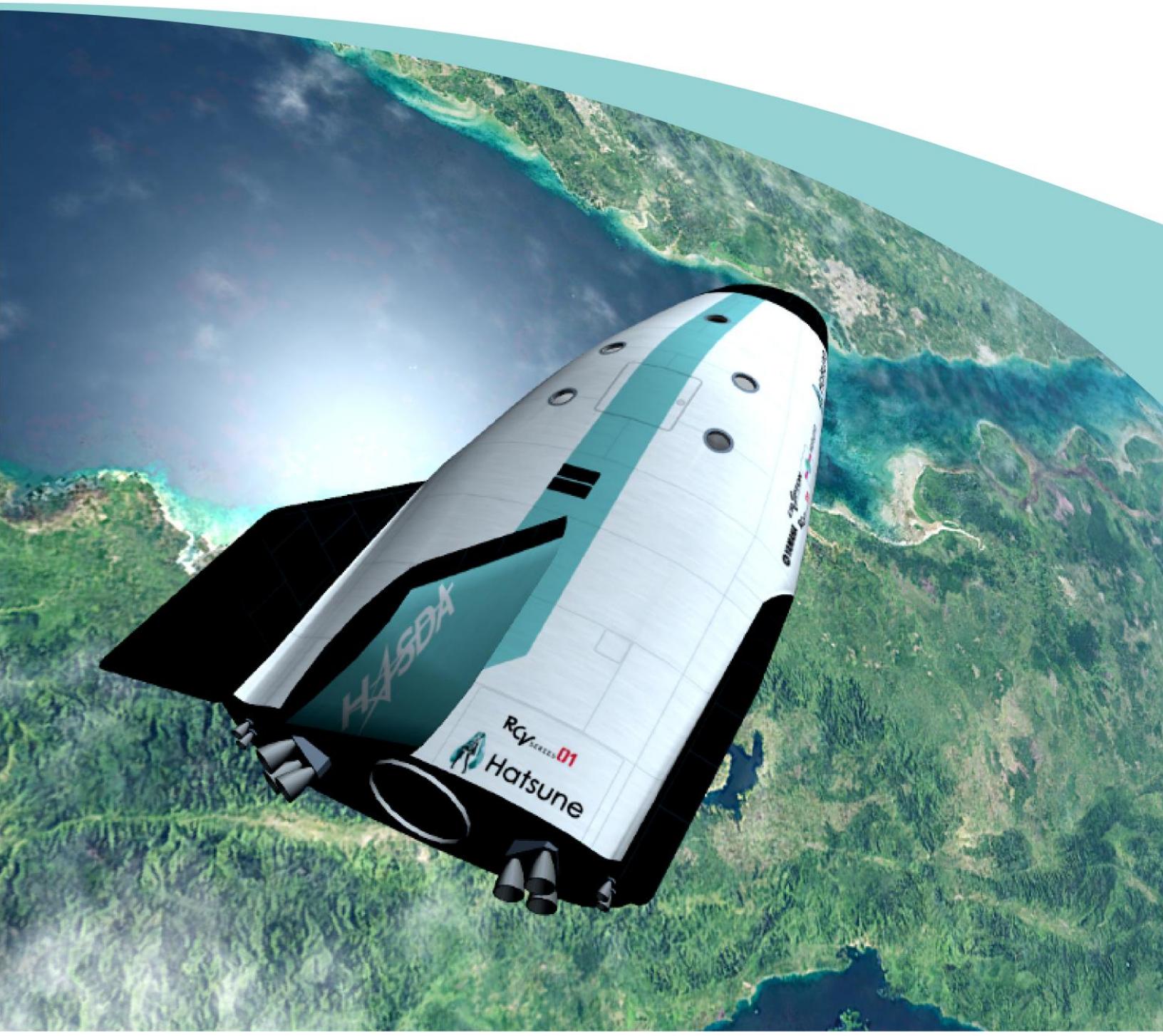
HASDA Hatsunia Aerospace Science & Development Agency

Reusable Crew Vehicle



v0.1.1 User Manual

Disclaimer

Hatsunia is a fictional, alternate version of Japan based on the virtual singing character "Hatsune Miku." The Hatsunese space program is a more ambitious version of the Japanese Aerospace Exploration Agency (JAXA). It is not unprecedented for Miku to be associated with space flight by her fans: in 2009, there was a successful 14,000-signature petition (which was also backed by a JAXA employee) to place images of Miku inside the Japanese Venus probe "Akatsuki," which was launched in 2010.
http://en.wikipedia.org/wiki/Hatsune_Miku#Cultural_impact

Crypton Future Aerospace is a fictional counterpart of Crypton Future Media (producers of the Miku singing synthesizer software). The Yamaha Corporation (producers of the Vocaloid software engine) in real life does not manufacture spacecraft components.

The RCV itself is based on an experimental JAXA aircraft known as LIFLEX (Lifting Body Flight Experiment). The RCV logo, initialism, and names are based on Crypton Future Media's "Character Vocal" (CV) series.

http://www.aero.jaxa.jp/eng/publication/magazine/sora/2007_no18/ss2007no18_01.html
http://www.crypton.co.jp/mp/pages/prod/vocaloid/cv01.jsp

Overview

The Reusable Crew Vehicle is a Hatsunese manned lifting body space plane. Since 2007, the Hatsunia Aerospace Science and Development Agency (HASDA) has been using the RCV to transport astronauts into low Earth orbit and back. It is launched on the M-II 01 launch vehicle and lands on a runway after a low-G re-entry. An expendable adapter is used for docking with space stations. The maximum passenger capacity of the RCV is five passengers, or three passengers with a few hundred kilograms of supplies. HASDA currently maintains a fleet of three RCVs, known as the "RCV Series":

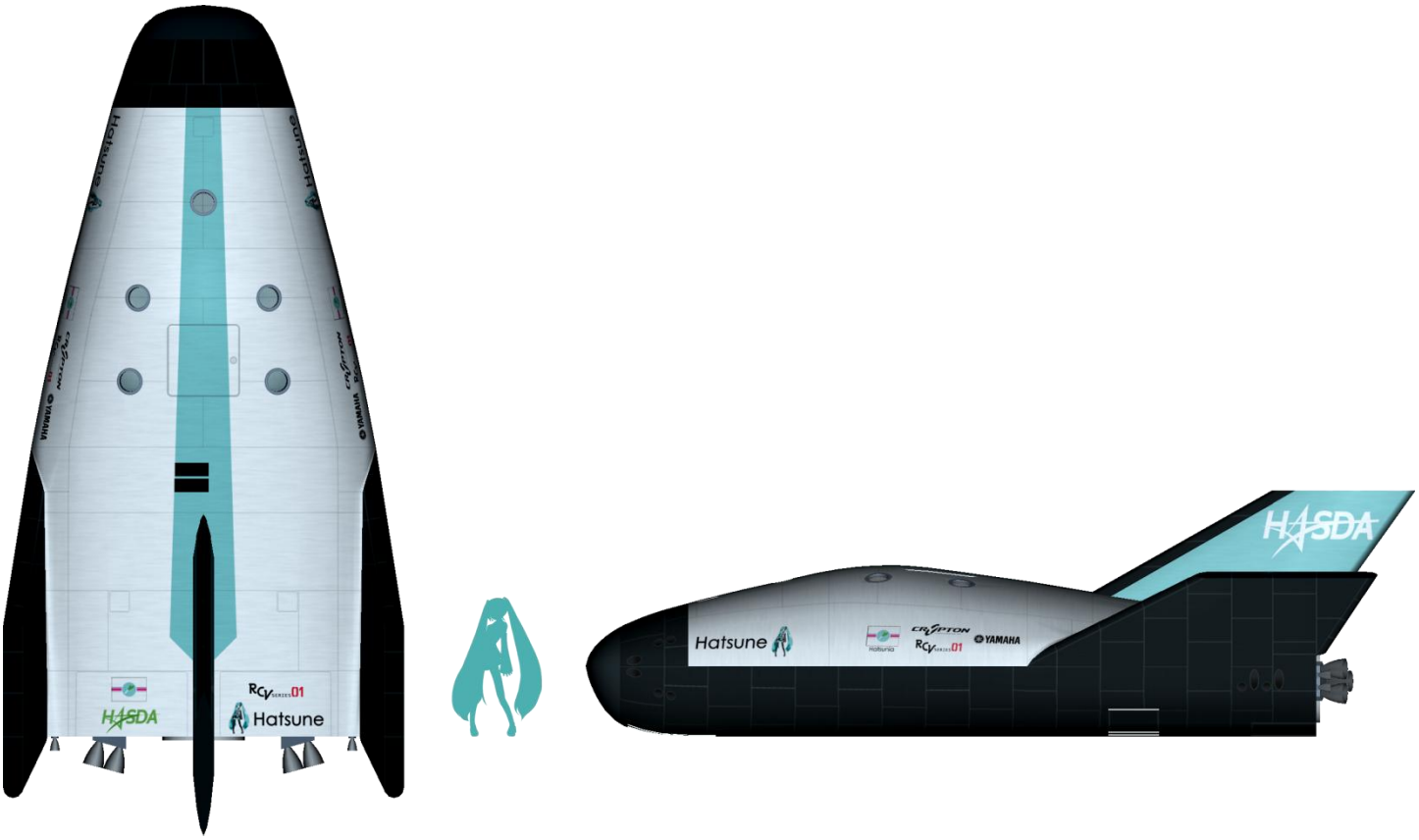
- RCV01 "Hatsune"
- RCV02 "Kagamine"
- RCV03 "Megurine"

There is also a NASA-owned RCV named "Spica." It is launched on the Atlas V 402 rocket.

The RCV Series

Installation procedure

- Most launch scenarios (except for RCV Spica) require **Negishima Space Center** (http://www.orbithangar.com/searchid.php?ID=6424) and the **M-II + Negi-5 launch vehicles** (http://www.orbithangar.com/searchid.php?ID=6438) along with **Velcro Rockets** (http://www.orbithangar.com/searchid.php?ID=3388). Download the .zip files and extract them to your Orbiter folder (where Orbiter.exe is located)
- For the RCV Spica launch scenario, sputnik's **Velcro EELVs** (http://www.orbithangar.com/searchid.php?ID=3462) is required for the Atlas V, and BrianJ's **MRO** (http://www.orbithangar.com/searchid.php?ID=3711) is required for the launch pad.
- This add-on requires the **spacecraft3** plug-in by Vinka. It is recommended that you download and install the compatible plug-in **genericvessel** (http://www.orbiter-forum.com/showthread.php?t=30362) as Vinka's original download page is not available anymore.
- AeroBrakeMFD (http://www.orbithangar.com/searchid.php?ID=2139) is required for re-entry.
- Extract this .zip file to your Orbiter folder. Scenarios are available in "Reusable Crew Vehicle".



Size comparison with Hatsune Miku (1.58 m)

Specifications

Crew	1 pilot and up to 4 passengers
Length	9.39 m (9.73 m with adapter)
Span	4.539 m
Empty mass	7390 kg
Propellant mass	1149 kg
Full mass	8539 kg
Adapter mass	600 kg
Total mass	9139 kg
Engines	6 x 3.9 kN engines (23.4 kN)
Propellant	N_2O_4/MMH
Specific impulse	316 s
Burn time	152 s
Delta-v	447.8 m/s (416.4 m/s with adapter)

Custom key controls

For RCV:

[G] – landing gear; [K] – speedbrake
[Shift] and [Numpad 1] – open/close rear hatch
[Shift] and [Numpad 2] – open/close top hatch

For docking adapter:

[K] - open/close hatch

For crew access tower (in M-II RCV launch scenarios):

[K] – rotate access arm

(Static/Non-functional) Virtual Cockpit

This virtual cockpit is purely decorational, for now. The left MFD has a HUD overlay that is designed to be used with CameraMFD (http://www.orbithangar.com/searchid.php?ID=2645), because the RCV has no forward facing windows. Press [Ctrl] + [H] to toggle the HUD on and off.

For practical purposes, use the "generic glass cockpit" mode ([F8] to switch modes), especially when the CameraMFD has a bad framerate in virtual cockpit mode.

Launch procedure

To launch into orbit, hold [Numpad +] and [Ctrl]. Once the rocket has cleared the launch tower, press [O] to activate the Velcro Rockets autopilot, which will continue until the spacecraft has reached orbit. Then, press [J] to detach.

Re-entry and landing procedure

Using the Map MFD, wait until your orbit crosses the desired landing site. Then turn retrograde and fire the engines half an orbit from the target until the periapsis is about 0 km altitude (6371 km radius from Earth's center).

(If the RCV has a docking adapter, detach by pressing [Left Shift] + [A], then [Left Shift] + [Numpad 0])

Rotate prograde, and wait until your altitude is getting close to 120 km. Rotate so that you are level with the horizon, then rotate up until the AOA (Angle of Attack) is almost 40 degrees.

Use AeroBrakeMFD (http://www.orbithangar.com/searchid.php?ID=2139) to show your predicted landing site. Click TGT to select the site, click PG until the top left corner says "Graph/Map", and click PRJ until you see a map. Use the rotational RCS mode gently so that the predicted landing site is as close as possible to the target (hold ctrl when tapping the numpad keys for more precision). When you get below 90-80 kilometers, lightly tap the trim keys ([Insert] and [Delete]) to adjust your trajectory.

Once the RCV is around Mach 2, and if the projected landing site is still close to the target, pitch down and start gliding to the runway. Use the speedbrake [K] to slow down if the vehicle is too fast, and deploy the landing gear with [G]. Ideal landing speed should be around 120-130 m/s, but don't slow too much or else there will be a stall.

Version history

- v0.1 – Initial release
- v0.1.1 – fixed missing docking adapter .dll + vessel .cfg

by Pipcard (mikusingularity) - 2015

Thanks to Martin Schweiger for creating Orbiter, sputnik for creating Velcro Rockets, Vinka for creating Spacecraft3, Artlav for creating the Spacecraft3 to DLL&C++ converter (for the docking adapter only, because the converter doesn't preserve exhaust textures), Hielor for creating the Wings3D .msh exporter, and the Orbiter community for helping me in creating this.