Kerbal Space Program - Bug #23003

(BaseServo PartModules) KSPFields showing current servo position aren't usable by mods

06/25/2019 05:48 PM - Dunbaratu

Status:	Ready to Test		**Start date:**	06/25/2019
Severity:	Low		**% Done:**	80%
Assignee:				
Category:	Plugins/Add-Ons			
Target version:	1.12.0			
Version:	1.7.2		**Language:**	English (US)
Platform:	Windows		**Mod Related:**	No
Expansion:	Breaking Ground			

Description

(This is affecting the kOS mod, but perhaps other similar mods like kRPC and Mechjeb might want this changed too.)

(**I am making this issue at the advice of SQUAD staff member TriggerAU with whom
I've exchanged a few posts about this on the KSP forums.**)

There exist some informational fields on the PAW panels for the new BaseServo PartModules that come with Breaking Ground DLC which are not being updated if the PAW is closed. This makes it hard for a mod like kOS to obtain the information it needs to "see" where these servo parts are currently set.

Currently, these fields are string values (which is its own kind of problem for kOS as it means a script needs to spend time parsing the string back into its numeric value to make use of it), but more importantly is that to save CPU time, KSP does not update these values if the PAW is closed, which means a kOS script can't read them properly as they get "frozen" at whatever value they previously showed the last time the PAW was open. (or empty string if the PAW was never opened yet since the last scene load).

I have built the following list of all such fields I could find from experimentation:

I give the names as they appear in the GUI PAW panel, which may or may not be the names they have under the hood in the actual KSPFields:

For ModuleRoboticServoPiston, these field names:

- "Current Extension" - when setting KSPfield "Target Extension", the piston does move, but the value reported by "Current Extension" doesn't change until the PAW is opened.

- "power" - when setting KSPField "Target Extension", the piston does move, which does draw power, but the value to "power" doesn't say so unless the PAW is open while the piston is moving. This field is also unusual in that it combines two different pieces of information together into the string - the current power draw and the max hypothetical power draw are both in the one string value.

For ModuleRoboticServoHinge, these field names:

- "Current Angle" - (same as above for ModuleRoboticServoPiston's "Current Extension".)

- "power" - (Same as above for ModuleRoboticServoPiston.)

For ModuleRoboticRotationServo, these field names:

- "Current Angle" - (same as above for ModuleRoboticServoPiston's "Current Extension".)

- "power" - (Same as above for ModuleRoboticServoPiston.)

For ModuleRoboticServoRotor, these field names:

- "Current RPM" - (same as above for ModuleRoboticServoPiston's "Current Extension".)

- "power" - (Same as above for ModuleRoboticServoPiston.)

History

#1 - 06/25/2019 11:29 PM - TriggerAu

- Status changed from New to Confirmed

- % Done changed from 0 to 10

#3 - 06/28/2019 05:56 PM - Dunbaratu

I occurs to me I should have called this "Feedback" rather than "Bug". "Bug" sounds a bit harsh because from the point of view of stock-only gameplay, everything is fine. This only causes a problem for mods trying to make use of these fields.

But I don't know how Squad's internal tracking system works so I don't know if changing it to "feedback" now will confuse things, so I will leave it as-is for now.

#4 - 07/04/2019 10:31 AM - TriggerAu

Well have a fix for the currentExtension/angle etc soon, but the "power" one may need some other stuff to settle first

#5 - 07/12/2019 07:25 AM - Dunbaratu

TriggerAu wrote:

> Well have a fix for the currentExtension/angle etc soon, but the "power" one may need some other stuff to settle first

I don't know if it was intended to fix it yet or not, but the latest KSP 1.7.3 didn't fix this. It **did** change things like "current angle" from a string to a single-precision float, but that floating point number still never updates if the hinge moves while the PAW is closed. (i.e. if the hinge moves because a kOS script wrote a new value to "target angle" rather than because a user clicked on "target angle" in the PAW, then the KSPField "current angle" still remains unchanged, even though the hinge did indeed move just fine. I just can't have a script query its current position because its current position field is wrong until the PAW is opened by a user.

If there was some way for kOS to just tell KSP "I changed a value, so the PAW is dirty and needs updating", that would be fine. Anything to get KSP to know it needs to update those fields even though the user didn't have the GUI up.

#6 - 01/14/2020 08:26 AM - Dunbaratu

This has never really been fixed. It's still a problem in KSP 1.8.1, and as far as I can tell there is no possible workaround I can do from my end. Nothing in the API that I can see seems to be able to make it work (which only includes public access members - if there's any workaround that involves calling something private or protected then I can't see it or call it, of course).

The essential problem is that the following members of the servo partmodules are wrong if the PAW is currently not open. As far as I can tell, whatever KSP does to update these fields, it only runs that update if the PAW is open. It may have some logic to skip around it when the PAW is closed, under the notion that if the user can't see it, the fact that the value is wrong doesn't matter. But it does affect mods trying to query this public field of the PartModule to find out what position the robotic part is currently in.

The fields in question are, again:

```
ModuleRoboticRotationServo.currentAngle
ModuleRoboticServoHinge.currentAngle
ModuleRoboticServoRotor.currentRPM
```

Surprisingly, this next one, which I'd think would follow the same pattern, does not. The following one actually does seem to update when the PAW is closed:

```
ModuleRoboticServoPiston.currentExtension
```

#7 - 02/15/2020 04:56 PM - victorr

- Status changed from Confirmed to Ready to Test

- Target version set to 1.9.0

- % Done changed from 10 to 80

We have made changes in this last release and would like your feedback please.

#8 - 02/16/2020 01:21 AM - Dunbaratu

- Status changed from Ready to Test to Not Fixed

- % Done changed from 80 to 50

victorr wrote:

> We have made changes in this last release and would like your feedback please.

I see no difference in the behavior from before.

I just recompiled kOS for KSP 1.9 just to be sure it wasn't that, and tried again, and still I see no difference in the behavior from before.

Exactly the same as before:

- ModuleRoboticServoPiston.currentExtension works. That field changes as soon as the part moves, whether the PAW is open or not. This is fine.

But these other three robotic servo types' fields that describe their current position do not work:

- ModuleRoboticRotationServo.currentAngle
- ModuleRoboticServoHinge.currentAngle
- ModuleRoboticServoRotor.currentRPM

If their PAW (part action window) is closed, they never update to new values. If I leave their PAW open over on the side of the screen somewhere, they do update.

Whatever is updating these fields of the PartModules only seems to be getting executed when the Part Action Window is open.

(The fact that the same problem does **not** happen with the piston implies the piston part is implemented a little differently.)

Is this (the fact that it only updates when the GUI PAW is open) accidental or deliberate? If updating those fields is deliberately avoided when the PAW is closed (perhaps it's expensive?), I could also be happy with a solution where there was some kind of public method I could call that makes the servo PartModule update the field, which I would call only when I see there's a need for it because a script is trying to query the value.

(Also, if there is a reason this is just not worth it and you want to close the issue unfixed, I can accept that too.)

#9 - 06/24/2021 10:49 PM - victorr

- Status changed from Not Fixed to Ready to Test

- Target version changed from 1.9.0 to 1.12.0

- % Done changed from 50 to 80

We have made some changes in this last 1.12.0 release and would like some feedback on this issue. Thanks.

#10 - 09/01/2021 05:06 AM - jj

no change