Replicant - Feature #1779

Implement setting that allows to permanently disable the modem

03/06/2017 10:06 PM - Wolfgang Wiedmeyer

Status:	New	**Start date:**	05/20/2017
Priority:	Normal	**Due date:**	
Assignee:	Fil Bergamo	**% Done:**	0%
Category:	Telephony and mobile data	**Estimated time:**	0.00 hour
Target version:	Replicant 6.0 0005	**Spent time:**	0.00 hour
Resolution:		**Grant:**	
Device:		**Type of work:**	Any programming languages (scripts, C, etc), Communication (mails, contacting people, etc), Wiki editions

Description

Some users don't need or don't want to use the modem of their device. Reasons might include skepticism about the level of the modem isolation, the wish to completely avoid the tracking of the mobile system or they simply don't want to have a nonfree system like the modem operating system running on their device.

So far, these users were advised to either buy a Replicant-supported device without a modem or to disable the radio interface layer by deleting or moving the library.

A more user-friendly approach would be a setting that, when enabled, would disable the modem boot when booting Replicant and that would not load the modem software to the modem.

When enabling the setting, the phone needs to be rebooted to ensure that the modem is not running. When disabling the setting, the user needs to be informed that the modem will only be operational after the device is rebooted.

Subtasks:

Feature # 1806: modem shell enable/disable script: verify with measurements through a B...	**New**

History

#1 - 03/06/2017 10:07 PM - Wolfgang Wiedmeyer

- Description updated

#2 - 04/22/2017 06:51 PM - Wolfgang Wiedmeyer

- Target version set to Replicant 6.0

#3 - 04/22/2017 07:06 PM - Wolfgang Wiedmeyer

- Assignee deleted (Paul Kocialkowski)

#4 - 05/19/2017 03:38 PM - Jeremy Rand

If /system/lib/libsamsung-ril.so is restored after booting the phone with the modem disabled, is it feasible to restart whatever process normally links to that .so file, in order to initialize the modem without rebooting the entire phone?

#5 - 05/20/2017 07:09 PM - Fil Bergamo

- Assignee set to Fil Bergamo

I can look into implementing a switch or so under the "system settings".
This is certainly doable.
But I will need more info on what that switch would need to do, actually (e.g. issuing a shell command? delete/rename a file? whatever else?)
For the moment, I can take the assignment of this issue, but I can't promise any working resolution.
I will research on that and update here with my findings.

#6 - 05/25/2017 01:59 PM - Wolfgang Wiedmeyer

Jeremy Rand wrote:

> If /system/lib/libsamsung-ril.so is restored after booting the phone with the modem disabled, is it feasible to restart whatever process normally links to that .so file, in order to initialize the modem without rebooting the entire phone?

It's worth a shot. RILD is the process to start and it should initialize the modem, but I'm not sure if the SIM card is then properly registered with Android if it wasn't at boot.

#7 - 05/28/2017 04:27 PM - Jeremy Rand

Wolfgang Wiedmeyer wrote:

> Jeremy Rand wrote:
>
> > If /system/lib/libsamsung-ril.so is restored after booting the phone with the modem disabled, is it feasible to restart whatever process normally links to that .so file, in order to initialize the modem without rebooting the entire phone?
>
> It's worth a shot. RILD is the process to start and it should initialize the modem, but I'm not sure if the SIM card is then properly registered with Android if it wasn't at boot.

Good news! I didn't even have to restart RILD myself. About 5-10 seconds after I restored /system/lib/libsamsung-ril.so to its original location, Replicant discovered the mobile network, including the SIM. I infer that RILD periodically checks for /system/lib/libsamsung-ril.so if it fails to find it on initial boot.

This is on the Galaxy S3 i9300 with Replicant 6.0 0001.

#8 - 08/31/2017 12:30 PM - Fil Bergamo

My understanding is that Jeremy's scripts are considered to be a good-enough-for-now solution.
Should I go on with my original intention, then?

If you think it's worth doing, I can come up with one of these options:

1. implement a standalone graphical app, wrapping Jeremy's scripts, quite like with RepWifi
2. include the option to disable RIL directly inside RepWifi
3. patch the System Settings app, adding a checkbox that executes the scripts under the hood when checked/unchecked.

Or.. Should we wait for a deeper understanding of modem (in)operation before going on?
Should this be the case, I think that what reported on #1820 is the way to go.. I'll post what I know on that page.

Please let me know what you consider to be the best choice!

Cheers,

Fil

#9 - 08/31/2017 07:08 PM - Wolfgang Wiedmeyer

Fil Bergamo wrote:

> My understanding is that Jeremy's scripts are considered to be a good-enough-for-now solution.
> Should I go on with my original intention, then?

In terms of usability and device-independence, there is still room for improvement ;)

> If you think it's worth doing, I can come up with one of these options:
>
> 1. implement a standalone graphical app, wrapping Jeremy's scripts, quite like with RepWifi
> 2. include the option to disable RIL directly inside RepWifi

Not sure if that's worth it. I think it's better to integrate it into the settings app.

> 1. patch the System Settings app, adding a checkbox that executes the scripts under the hood when checked/unchecked.

If you already do the whole work of adding the settings there, it should be based on a proper solution that is as device independent as possible. The scripts certainly aren't and what goes into the settings app should not be specific for a certain hardware interface implementation. Then it's a lot easier to account for devices that don't have a modem or that don't use Samsung-RIL. Otherwise, it becomes a pile of hacks over time.

Using the approach with the RIL daemon I described, it's also not entirely sure that it will work for all sorts of possible hardware that can be supported by Replicant. There is also the case that a dedicated daemon is used to boot the modem and the vendor RIL does the rest. The approach would not help in this case. I currently have it this way for QMI-RIL. But I'm planning to change that and also let the RIL control the modem boot.

> Or.. Should we wait for a deeper understanding of modem (in)operation before going on?
> Should this be the case, I think that what reported on #1820 is the way to go.. I'll post what I know on that page.

Testing it with such a cell would certainly help a lot to ensure that it works as intended. Now, there is not really a way to test it.

#10 - 08/30/2018 05:03 PM - Kurtis Hanna

- Target version changed from Replicant 6.0 to Replicant 6.0 0005

#11 - 12/29/2018 10:30 AM - Fil Bergamo

I really want to work on this to provide the front-end for disabling the modem.
But I will need some strong solutions in the back-end to work with.

Has anybody looked into this in details?
What would be a general way of unloading the modem firmware that works for every supported device?
Do every device work the same, in that a firmware binary resides in a certain filesystem location?
Is moving that binary away an equivalent workaround for all devices?

I can **really** work on the settings app efficiently if anybody is able to answer those questions.

#12 - 03/23/2021 11:28 AM - _I3^ RELATIVISM

wouldnt a approach like postmarket os, to give this feauture during installation a easier solution for this? Or the idea is to be switchable at all times?

#13 - 03/23/2021 11:28 AM - _I3^ RELATIVISM

- Type of work Any programming languages (scripts, C, etc), Communication (mails, contacting people, etc), Wiki editions added