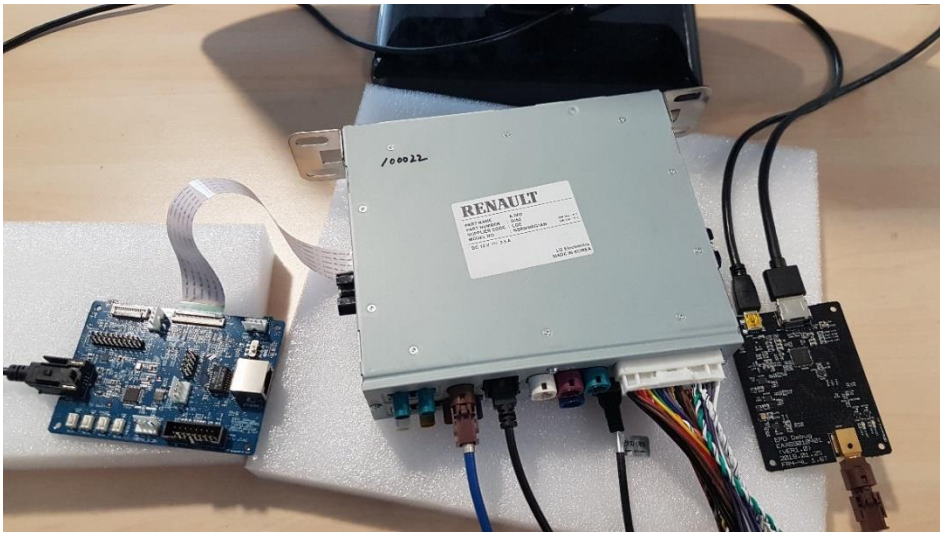
Report of the Week

a) Introduction

In order to prepare the Demo canoe with the *AVM* for *Parkassist*, it was necessary for this week to reflash my LG board by going from *p.car* to *android Q*. To do this, the idea was to create my own build on the server before proceeding to the flash of the board.



b) Build creation

To create the build, I connected to the server and did the *repo init* command after creating a new directory. This creates a *.repo/* directory with Git repositories for the Repo source code and the standard Android manifest files. So as it was for the *android Q*, I took the following repo to to achieve this goal

https://gitlabee.dt.renault.com/swlabs/cc/ivi/android/product/loire/manifest -b release_aivi2_03.05.30.Q0M.sit -m default.xml.

After this, I started the *repo sync -j24 -f* command to synchronize the files for all project as well as to finish with the *source*, *lunch* and the *make dist* commands. For the *lunch,* I take the *avivi2_full-userdebug* as on the following capture

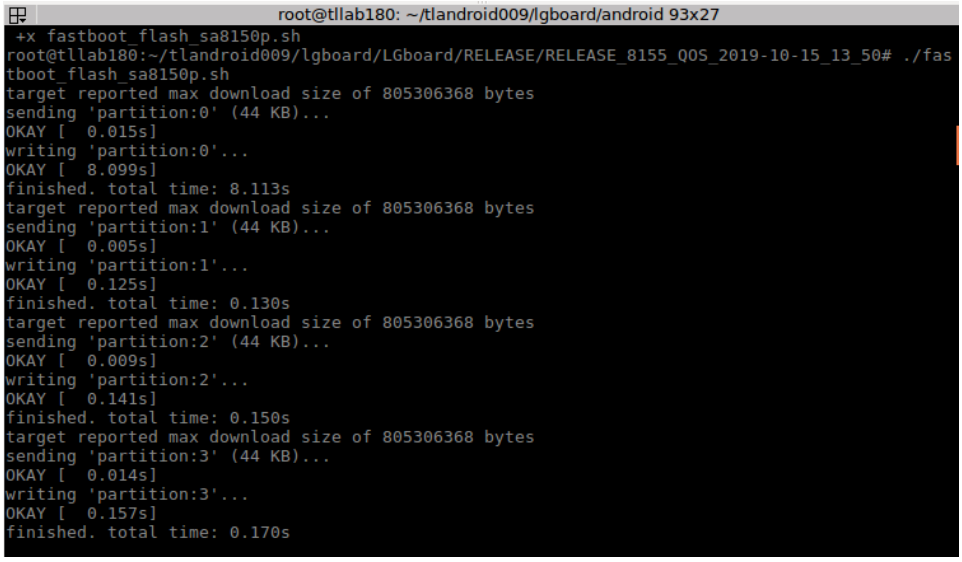
```
root@tllab180: ~/tlandroid009/lgboard/android 93x27
root@tllab180:~/tlandroid009/lgboard/android# source build/envsetup.sh
including device/alliance/aasp/vendorsetup.sh
including device/alliance/aivi2_full/vendorsetup.sh
including device/generic/car/vendorsetup.sh
including device/generic/mini-emulator-arm64/vendorsetup.sh
including device/generic/mini-emulator-armv7-a-neon/vendorsetup.sh
including device/generic/mini-emulator-x86_64/vendorsetup.sh
including device/generic/mini-emulator-x86/vendorsetup.sh
including device/generic/uml/vendorsetup.sh
including device/qcom/common/vendorsetup.sh
including device/qcom/qssi/vendorsetup.sh
including vendor/qcom/opensource/core-utils/vendorsetup.sh
including vendor/qcom/proprietary/common/vendorsetup.sh
including sdk/bash_completion/adb.bash
root@tllab180:~/tlandroid009/lgboard/android# lunch

You're building on Linux

Lunch menu... pick a combo:
     1. aosp_arm-eng
     2. aosp_arm64-eng
     3. aosp_mips-eng
     4. aosp_mips64-eng
     5. aosp_x86-eng
     6. aosp_x86_64-eng
     7. aasp_emulator-userdebug
     8. aivi2_full-userdebug
```

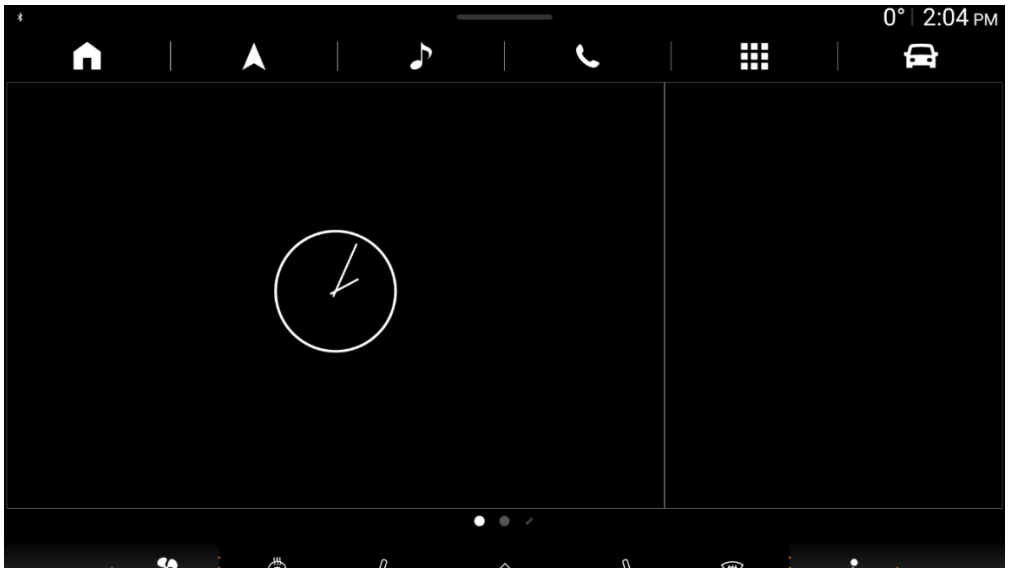
c) Flashing Board

First, I created a *realese* with the build I had just created by running the file *realize_image_sa8150.sh* and then retrieving it locally by *sshfs* and finally running the *fastboot* command in local after having done *adb reboot bootloader*. So the following capture shows the end of the board's flash.

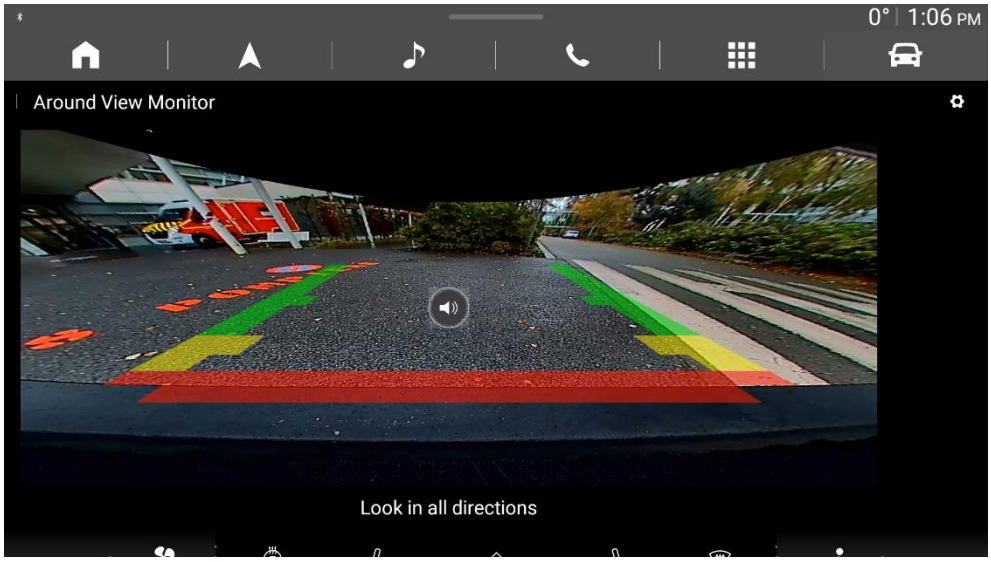


d) board flash result

After finishing the board flash, I got the following results on the screen that connected to the board by HDMI



So, with this result obtained we have access to Parkassist, as on the following capture, which is the subject of my next demo.



e) For the next week

As previously announced, it is a question of reproducing for next week the demo of *Micha Soares* with the 4 cameras connected to the board to be able to recover not only the images of the cameras but also to change the view with buttons via the *Parkassit* application seen on the last capture.