GNSS high accuracy positioning split arch

ACU6-PRO SDK

Installation

The setup guide can be found on https://production.connect.actia.se/. From there, you can download the latest version which also contains HTML links for the setup.

The environment requires Ubuntu. On windows, WSL will not work if using Windows 10 as it isn't USB compatible. Could possibly work with Windows 11. VirtualBox works fine though with the latest versions of Ubuntu. On VirtualBox, you need to allow USB 3.0 which can be easily done from settings. The Punchboot only showed up when I tried to flash. I had to allow the USB to be used as soon as I flashed the firmware the first time. Could be done from the menu bar on the virtual machine. Otherwise, it wouldn't write to the SDK.

For me, docker didn't work if I downloaded Docker Desktop on my virtual machine but I worked fine if I only used Docker engine but could be a fault from my side.

Help: acu6.support@actia.com helped a lot with set up, if there comes up any uncertainties.

Buildroot

Create new packages

To create a new package, you must create a folder in the *external/myproj/package*. The folder should include a *<package_name>.mk* file and a *Config.in* file. The config in file is for dependencies and *.mk* file is for the build. *<PACKAGE_NAME>_BUILD_CMDS* is for the build commands that need to be ran and *<PACKAGE_NAME>_INSTALL_TARGET_CMDS* for how it should be installed. These two are the main ones and the only I needed to use for my projects. Check the different packages included in the git repository for inspiration.

In the */externals/myproj/Config.in* file, you must add the path to your *Config.in* file in your new package.
In the */externals/myproj/configs/myproj_defconfig* file, you also have to add the package name manually, the same way they have done it.

Actia-lib

The Actia library is used to get cell data, a connection, imsi and more. In the *.mk* file, you need to specify that Actia libary is a dependency like this:
<PACKAGE_NAME>_DEPENDENCIES = actia-lib

And the *Config.In* file needs the following line:
select BR2_PACKAGE_ACTIA_LIB

In the *Makefile* for the project, you also need to add -lactia on the LIBS row:
LIBS += -lactia

If using cmake: In *CMakeLists.txt*, the one placed in the same folder as the files using *actia-libary*, the following should be added instead:
target_link_libraries(example PRIVATE actia)

Building

Only run make from the bottom of the directory. Do not build the files in their folder in *external/…/package*. Otherwise, it isn't built for the correct machine.

Lastly, *make flash* will move the package to the ACU6-PRO.

Scripts

Check package *runscript* to see how you build scripts. To create a new script, you need to create a new package with a new *.sh* file. Can copy the *runscript* folder and rename it to do so. It might be possible to add a script to the same package and make some changes to the *runscript.mk* file.

When SSH onto the acu6pro. The scripts will be placed in *../usr/bin.* For some reason, nothing shows up when running ls so first one must run *cd ..* to be able to se all the folders on the ACU6.

SUPL-3GPP-LPP-Client

The SUPL-3GPP-LPP-client (v3.2.0) can be run with the *runscript.sh* file. The serial port it uses to communicate with the ublox is rs232. This is done with the line *--serial /dev/rs232* as in the *runscript.sh* file.

To be able to run the client, you first have to request a connection and set the baudrate correct. The *runscript.sh* does all of that. It sets the cell though and the cell and more, which the ACU6PRO should be able to change.

Right now, the client is pulled from git and to change version, change the version in the .mk file and rebuild. The git includes to correct version *of SUPL-client* but make sure that the *.mk* file points to the correct repository and version. It is possible to have the program locally instead of pulling from git but it in currently not implemented.

In the SUPL-Client v3.2.0, in *osr_example.cpp*, *beidou* should be set to false for correct compatibility with u-center. The version of the SUPL-Client on the git for this has that.

Get cell data + imsi (my-modem)

My-modem.c and *my-modem.h* includes files used to return imsi and fetch relevant cell data. The files are in *src/* in the SUPL-client. The functions to be called are commented and included in the header-file (*my-modem.g*).

The following function returns the imsi as an *unsigned long*:
get_imsi();

The following function returns the cell data from ACU6-PRO as a *struct CellID* (from lpp/*cell_id.h*):
get_cell_data();

The *get_cell_data()* is run on a separate thread inside the *osr_example.cpp*. Improvement that could be made would be to call *subscribe* ones and then only call for *wait_for_status* with some modifications (in *my-modem.c*).

A version of the code for *my-modem* also exists in a separate package used to test function. It currently includes some previous functions that was used before the updated ones in the one placed in the SUPL-client.

For more inspiration on how to get cell data, check out the demo files, especially *actia-modemcontrol-demo* in the */sdk/acu6…/externals/acu6-demos/package* folder.

U-Blox

The SUPL-client runs on the ACU6-PRO and communicates with the ublox though the serial port rs232 (yellow one on our version of ACU6-PRO). The baudrate needs to be set correct, *runscript.sh* does that, with this command when SSH on the ACU6:
stty -F /dev/rs232 115200 cs8

We used a ublox C102-F9R-0-00 which connects to the ACU6 with rs232 and an antenna. A computer with u-center can connect to it.

On u-center, on *view->messages* view. Go to *UBX->NAV->SIG* and enable messages and you can see relevant data.

Set Up from the git version

The git contains a folder named *externals*. That should replace the *externals* package when the installation of the ACU6-PRO is fully completed. The content of the SUPL-client folder should be pushed to a git and then pulled from it if not set up locally. The *.mk* file (*externals/myproj/package/supl-3gpp-lpp-client*) for the SUPL-client should be edited so it pulls the correct version. Change version and source, so it points correctly. Recommend trying to set it up locally though, otherwise you must push changes to git and change the version in the *.mk* file for every change.

Problems

I have noticed that the client sometimes crashes but very seldom. And sometimes, it seems that the cell from acu6 on startup isn't the correct one. This seems to happen if you run the SUPL-client very quickly after restarting the ACU6-PRO but if the client is restarted, it usually works correctly. I have not been able to find the cause.

Error message:

```
terminate called after throwing an instance of 'std::runtime_error'
  what():  Unable to process LPP client (probably disconnected)
Aborted
```