**### BUILDING ROS on GLSDK with YOCTO PROJECT ###**

1. Make sure that GLSDK is installed on host machine.
2. Setup the GLSDK environment variable to the location where the GLSDK is installed.

**host $** export GLSDK="/path/to/glsdk/installation/ti-glsdk\_dra7xx-evm\_6\_10\_00\_02"

1. Execute the script in the GLSDK release directory using:

**host $** cd ${GLSDK}

**host $** ./setup.sh

1. Add the Linaro cross-compile toolchain path in the PATH environment variable.

**host $** export PATH=<*Path to Linaro cross-compile toolchain*>/bin:$PATH

1. Run this command to as a one-time setup for the yocto build

**host $** cd $GLSDK

**host $** ./bin/setup-yocto.sh

1. Create a downloads directory (if building using Yocto for the first time), where the Yocto build will place the downloads. Note the path of the directory.

**host $**  cd yocto-layers

**host $** mkdir downloads

1. Once the GLSDK setup is done and setup-yocto.sh is run, add meta-ros layer to configs file,

**host $** gedit ./configs/glsdk/glsdk-6.10.00.02.txt

and add the following line above meta-glsdk git repository link,

**meta-ros,git://github.com/bmwcarit/meta-ros.git,master,HEAD,layers=**

1. Add the ros packagegroups to the image recipe which is going to be build, (i.e arago-glsdk-multimedia-image.bb)

**host $**  gedit ./sources/meta-glsdk/meta-arago-distro/recipes-core/images/arago-glsdk-multimedia-image.bb

and append ros -packagegroup & gst packagegroup to IMAGE\_INSTALL as follow,

IMAGE\_INSTALL += “\

packagegroup-arago-base \

packagegroup-arago-console \

packagegroup-arago-glsdk-console \

packagegroup-arago-glsdk-multimedia \

packagegroup-arago-tisdk-connectivity \

packagegroup-arago-test \

packagegroup-arago-tisdk-matrix \

***packagegroup-arago-gst \***

***packagegroup-ros-comm \***

***packagegroup-ros-world \***

“

1. The packagegroup-ros-comm.bb and packagegroup-ros-world.bb recipes files are located at ./sources/meta-ros/recipes-ros/packagegroups. You can edit packagegroup-ros-world and remove the packages that are not needed from RDEPENDS\_${PN} variable.

**Host $** gedit ./sources/meta-ros/recipes-ros/packagegroups/packagegroup-ros-world.bb

and remove the following lines in RDEPENDS\_${PN} variable.,

*cyclic-timer-tests \*

*communication-tests \*

*oneshot-timer-tests \*

*ros-pocketsphinx \*

*sound-play \*

*gscam \*

1. Edit the recipe files collada-urdf\_1.10.20.bb and octomap-ros\_0.3.1.bb .

**Host $** gedit ./sources/meta-ros/recipes-ros/robot-model/collada-urdf\_1.10.20.bb

and replace the line ***${B}/CMakeFiles/$f/build.make*** with ***${B}/build/CMakeFiles/$f/build.make***

and save the file.

**Host $** gedit ./sources/meta-ros/recipes-ros/octomap-ros/octomap-ros\_0.3.1.bb

and replace the line ***${B}/CMakeFiles/$f/build.make*** with ***${B}/build/CMakeFiles/$f/build.make***

and save the file.

1. Once the ros-packagegroups are added we can run the build by running the script build-core-sdk.sh,

**host $** ./build-core-sdk.sh dra7xx-evm

1. This script will invoke Bitbake which will generate list of tasks and execute them. At the end the script will build the arago-glsdk-multimedia-image, and generated images can be found in ***yocto-layers/build/arago-tmp-external-linaro-toolchain/deploy/images/***
2. Copy the arago-glsdk-multimedia-image-<*MACHINE-NAME*>-<*DATE*>.rootfs.tar.gz to filesystem folder and run mksdboot.sh to prepare bootable sd card,

**host $**  cp ./build/arago-tmp-external-linaro-toolchain/deploy/images/ arago-glsdk-multimedia-image-<*MACHINE-NAME*>-<*DATE*>.rootfs.tar.gz ${GLSDK}/filesystem

\* Now connect SD card to PC

\* Make two partitions on the sd card boot(fat32) and rootfs (ext4).

\* Find out the SD card device name

**host $** sudo fdisk -l

\* Then run mksdboot.sh ,

**host $** sudo ${GLSDK}/bin/mksdboot.sh --device /dev/<device-name> --sdk ${GLSDK}

Eg: sudo ${GLSDK}/bin/mksdboot.sh --device /dev/sdc --sdk ${GLSDK}

14. After completion of step 13, eject the SD card, insert it on Vayu and reboot it.

**RUNNING ROSCORE ON VAYU:**

15**.** On Vayu login as root,

**target$ login:** root

16. Create a ros\_init script file in /etc/init.d and source ros\_init on every boot, this script will export ROS environment variables.

**target $** gedit /etc/init.d/ros\_init

and add the following lines.

## file: /etc/init.d/ros\_init

#!/bin/sh

export ROS\_ROOT=/opt/ros/hydro

export PATH=$PATH:/opt/ros/hydro/bin

export LD\_LIBRARY\_PATH=/opt/ros/hydro/lib

export PYTHONPATH=/opt/ros/hydro/lib/python2.7/site-packages:/usr/lib/python2.7/site-packages

export ROS\_IP="$(ifconfig | grep -A 1 'eth0' | tail -1 | cut -d ':' -f 2 | cut -d ' ' -f 1)"

export ROS\_MASTER\_URI=http://localhost:11311

export CMAKE\_PREFIX\_PATH=/opt/ros/hydro

touch /opt/ros/hydro/.catkin

## EOF

**target $** source /etc/init.d/ros\_init

17. Finally, you can start roscore with

**target $** roscore