GDP Master

- What is GDP Master?
- Releases and support
  - Generic Preparation
- Per-target instructions
  - QEMU x86_64
  - Renesas R-Car (Gen 3) M3 Starter Kit
  - Renesas R-Car (Gen 3) H3 Starter Kit
  - Renesas R-Car (Gen 3) M3 Salvator-X
  - Renesas R-Car (Gen 3) H3 Salvator-X
  - Raspberry Pi 2
  - Raspberry Pi 3
  - Intel Minnowboard Max/Turbot
  - Qualcomm DragonBoard 410c
  - Renesas R-Car M2 Porter (supported on GDP 11 but not GDP 12)
  - Renesas R-Car E2 Silk (supported on GDP 11 but not GDP 12)
  - Renesas R-Car M2 Koelsch (supported on GDP 11 but not GDP 12)
- What will I find in Master?
- Requests

What is GDP Master?

The GDP Master is essentially a rolling release that holds the layers, recipes, and source code to build the GENIVI Development Platform. It is the "latest and greatest" software that is the basis for the versioned releases happening twice a year. These versioned releases are supported from the date of release to following release. Changes go through basic build & run testing before ending up and the master branch, and we try to keep GDP Master relatively stable. It is therefore recommended to use Master for your development and trials on GDP. Most of the time this is your best choice!

Master is designed for:

1. Developers who want to develop automotive software components.
2. Developers who want to contribute to the GDP project itself.

Master policies can be summarized in these points:

1. There is one main branch on the repository, master, supporting all hardware targets.
2. We may use feature branches, temporarily, to prepare a larger work before merging to master.
3. Most things go to Master directly, after having been provided as a Pull Request, and tested.
4. We also use release/maintenance branches. The README and the next chapter clarifies the extent of backports.
5. Released versions are tagged. Remember that using master is most of the time recommended.

Releases and support

GDP is a community project and all adopters will get out of it what you put into it. Based on current contribution levels and contracted developers the following is the expected "support level" for processing bugs and change requests, and for backporting fixes to older versions. The main focus is on the latest Master development at any time, not on supporting older releases. With your community input, any of those things can be changed of course.

With the current resources we can only fully support the most recent full release plus GDP Master development, with critical bug fixes introduced also into the previous release.

In other words after release "n" has been officially released:

- GDP <n-1> – critical bugfixes only. Limited testing (those that still depend on n-1 need to support the testing efforts and report results)
- GDP <n> – supported
- GDP <n+1> – under current development = GDP Master

Generic Preparation

For all targets, the build needs the following preparation.

- Install the required development tools on the host by executing the following command (or equivalent on your distro. More information in Yocto Docs)

```
$ sudo apt-get install gawk wget git diffstat unzip texinfo gcc-multilib build-essential chrpath socat libstdc++-6-dev xterm
```

- Clone genivi demo platform
$ git clone https://github.com/GENIVI/genivi-dev-platform.git
$ cd genivi-dev-platform

Now execute per-target instructions below and then come back here:

- After executing bitbake of the image successfully, try your image.
- You can connect with SSH. You may need to figure out the IP address assigned via DHCP by using your network router, or scanning, etc.
- You can also usually connect with a serial link (see hardware specific instructions).
- Remember that the initial user / password is root / root

Per-target instructions

QEMU x86_64

For QEMU the build instructions are as follows:

Follow the generic instructions above. For the init step do the following:

$ cd genivi-dev-platform
$ source ./init.sh qemux86-64

Bitbake the image:

$ bitbake genivi-dev-platform

To try the image, follow the Hardware and Software Setup instructions.

Renesas R-Car (Gen 3) M3 Starter Kit

Follow the generic preparation instructions above. Once you reach the init step do the following.

At the moment, the following extra package also needs to be on your build machine:

$ sudo apt-get install ImageMagick

Proceed as follows:

- $ cd genivi-dev-platform
  $ source ./init.sh r-car-m3-starter-kit

- Obtain and Install Renesas Graphics Drivers

  Download the Click-through licensed Linux Drivers and Gfx/MMP packages for YBSP v3.9.0 and Wayland 1.14 / Weston 3.0 from the section for YP 2.5 (Sumo) from here and unzip them into a folder.

  $ cd <folder containing the two zip files>
  $ unzip -o R-Car_Gen3_Series_Evaluation_Software_Package_for_Linux-*.zip
  $ unzip -o R-Car_Gen3_Series_Evaluation_Software_Package_of_Linux_Drivers-*.zip

  To install them into the correct place in the Yocto BSP a copy script is used.

  $ cd ../meta-renesas

  If you have the click-through packages (no NDA):

  $ sh meta-rcar-gen3/docs/sample/copyscript/copy_evaproprietary_softwares.sh <path to the folder containing the packages>

  If you have the Evaluation packages (with NDA):

  $ sh meta-rcar-gen3/docs/sample/copyscript/copy_proprietary_softwares.sh <path to the folder containing the packages>

Bitbake the image:

$ bitbake genivi-dev-platform

- Once built, Setup Hardware & Deploy

Renesas R-Car (Gen 3) H3 Starter Kit
Follow the generic preparation instructions above. Once you reach the init step do the following.

At the moment, the following extra package also needs to be on your build machine:

$ sudo apt-get install ImageMagick

Proceed as follows:

- $ cd genivi-dev-platform
  $ source ./init.sh r-car-h3-starter-kit

- Obtain and Install Renesas Graphics Drivers

  Download the Click-through licensed Linux Drivers and Gfx/MMP packages for YBSP v3.9.0 and Wayland 1.14 / Weston 3.0 from the section for YP 2.5 (Sumo) from here and unzip them into a folder.

  $ cd <folder containing the two zip files>
  $ unzip -o R-Car_Gen3_Series_Evaluation_Software_Package_for_Linux-*.zip
  $ unzip -o R-Car_Gen3_Series_Evaluation_Software_Package_of_Linux_Drivers-*.zip

  To install them into the correct place in the Yocto BSP a copy script is used.

  $ cd ../meta-renesas

  If you have the click-through packages (no NDA):

  $ sh meta-rcar-gen3/docs/sample/copyscript/copy_evaproprietary_softwares.sh <path to the folder containing the packages>

  If you have the Evaluation packages (with NDA):

  $ sh meta-rcar-gen3/docs/sample/copyscript/copy_proprietary_softwares.sh <path to the folder containing the packages>

Bitbake the image:

$ bitbake genivi-dev-platform

- Once built, Setup Hardware & Deploy

Renesas R-Car (Gen 3) M3 Salvator-X

Follow the generic preparation instructions above. Once you reach the init step do the following.

At the moment, the following extra package also needs to be on your build machine:

$ apt-get install ImageMagick

- Initialise build environment

  $ cd genivi-dev-platform
  $ source ./init.sh r-car-m3-starter-kit

- Configure the build for M3 Salvator-X

  Until Salvator-X support is added to the GDP board template mechanism it is straightforward to reconfigure a M3 Starter Kit build (the target you passed as a parameter in the previous step) to build for M3 Salvator-X instead by editing the Yocto local.conf.

  To do that simply edit gdp-src-build/conf/local.conf to set the following MACHINE variable:

  MACHINE = "salvator-x"

- Obtain and Install Renesas Graphics Drivers

  Download the Click-through licensed Linux Drivers and Gfx/MMP packages for YBSP v3.9.0 and Wayland 1.14 / Weston 3.0 from the section for YP 2.5 (Sumo) from here and unzip them into a folder.

  $ cd <folder containing the two zip files>
  $ unzip -o R-Car_Gen3_Series_Evaluation_Software_Package_for_Linux-*.zip
  $ unzip -o R-Car_Gen3_Series_Evaluation_Software_Package_of_Linux_Drivers-*.zip

  To install them into the correct place in the Yocto BSP a copy script is used.

  $ cd ../meta-renesas

  If you have the click-through packages (no NDA):

  $ sh meta-rcar-gen3/docs/sample/copyscript/copy_evaproprietary_softwares.sh <path to the folder containing the packages>

  If you have the Evaluation packages (with NDA):

  $ sh meta-rcar-gen3/docs/sample/copyscript/copy_proprietary_softwares.sh <path to the folder containing the packages>
$ sh meta-rcar-gen3/docs/sample/copyscript/copy_evaproprietary_softwares.sh <path to the folder containing the packages>

If you have the Evaluation packages (with NDA):
$ sh meta-rcar-gen3/docs/sample/copyscript/copy_proprietary_softwares.sh <path to the folder containing the packages>

- Bitbake the image
  $ bitbake genivi-dev-platform

- Once built, Setup Hardware & Deploy

Renesas R-Car (Gen 3) H3 Salvator-X

Follow the generic preparation instructions above. Once you reach the init step do the following.

At the moment, the following extra package also needs to be on your build machine:

$ sudo apt-get install ImageMagick

- Initialise build environment
  $ cd genivi-dev-platform
  $ source ./init.sh r-car-h3-starter-kit

- Configure the build for H3 Salvator-X
  Until Salvator-X support is added to the GDP board template mechanism it is straight forward to reconfigure a H3 Starter Kit build (the target you passed as a parameter in the previous step) to build for H3 Salvator-X instead by editing the Yocto local.conf.

  To do that simply edit gdp-src-build/conf/local.conf to set the following MACHINE variable:

  MACHINE = "salvator-x"

- Obtain and Install Renesas Graphics Drivers

  Download the Click-through licensed Linux Drivers and Gfx/MMP packages for YBSP v3.9.0 and Wayland 1.14 / Weston 3.0 from the section for YP 2.5 (Sumo) from here and unzip them into a folder.

  $ cd <folder containing the two zip files>
  $ unzip -o R-Car_Gen3_Series_Evaluation_Software_Package_for_Linux-*.zip
  $ unzip -o R-Car_Gen3_Series_Evaluation_Software_Package_of_Linux_Drivers-*.zip

  To install them into the correct place in the Yocto BSP a copy script is used.

  $ cd ../meta-renesas

  If you have the click-through packages (no NDA):

  $ sh meta-rcar-gen3/docs/sample/copyscript/copy_evaproprietary_softwares.sh <path to the folder containing the packages>

  If you have the Evaluation packages (with NDA):

  $ sh meta-rcar-gen3/docs/sample/copyscript/copy_proprietary_softwares.sh <path to the folder containing the packages>

- Bitbake the image

  $ bitbake genivi-dev-platform

- Once built, Setup Hardware & Deploy

Raspberry Pi 2

Follow the generic preparation instructions above. Once you reach the init step do the following.

- Initialise build environment
  $ cd genivi-dev-platform
  $ source ./init.sh raspberrypi2

- Once built, Setup Hardware & Deploy
Raspberry Pi 3
Follow the generic preparation instructions above. Once you reach the init step do the following.

- Initialize build environment
  - $ cd genivi-dev-platform
  - $ source ./init.sh raspberrypi3
- $ bitbake genivi-dev-platform
- Once built, Setup Hardware & Deploy

Intel Minnowboard Max/Turbot
Follow the generic preparation instructions above. Once you reach the init step do the following.

- Initialize build environment
  - $ cd genivi-dev-platform
  - $ source ./init.sh minnowboard
- Bitbake the image
  - $ bitbake genivi-dev-platform
- Once built, Setup Hardware & Deploy

Qualcomm DragonBoard 410c
Follow the generic preparation instructions above. Once you reach the init step do the following.

- Initialize build environment, making sure to read and accept the terms of the EULA first.
  - $ cd genivi-dev-platform
  - $ source ./init.sh dragonboard-410c accept-eula
- Bitbake the image
  - $ bitbake genivi-dev-platform
- Once built, Setup Hardware & Deploy

Renesas R-Car M2 Porter (supported on GDP 11 but not GDP 12)
Follow the generic preparation instructions above. Once you reach the init step do the following.

- Initialize build environment
  - $ cd genivi-dev-platform
  - $ source ./init.sh porter
- Obtain and Install Renesas Graphics Drivers
- Bitbake the image
  - $ bitbake genivi-dev-platform
- Once built, Setup Hardware & Deploy

Renesas R-Car E2 Silk (supported on GDP 11 but not GDP 12)
Follow the generic preparation instructions above. Once you reach the init step do the following.

- Initialize build environment
  - $ cd genivi-dev-platform
  - $ source ./init.sh silk
- Obtain and Install Renesas Graphics Drivers
- Bitbake the image
  - $ bitbake genivi-dev-platform
- Once built, Setup Hardware & Deploy

Renesas R-Car M2 Koelsch (supported on GDP 11 but not GDP 12)
Follow the generic preparation instructions above. Once you reach the init step do the following.

- Initialise build environment
  
  ```bash
  $ cd genivi-dev-platform
  $ source ./init.sh koelsch
  ```

- Obtain and Install Renesas Graphics Drivers  NOTE that you must select "porter" for the graphics install step also for koelsch board.

- Correct meta-genivi-dev bug

  Apply the following patch

- Bitbake the image

  ```bash
  $ bitbake genivi-dev-platform
  ```

- Once built, Setup Hardware & Deploy

What will I find in Master?

GDP components are described in GDP in Detail wiki page.

Requests

Requests for the next GDP version, should be created as JIRA tickets in the GDP project, with the Feature type. It might be useful to first explain your proposal on the relevant mailing list

- GDP-154 - GDP Master requests [To do]. The requests will be processed and added first to GDP Master and then, according to the GDP policy, to the next release.