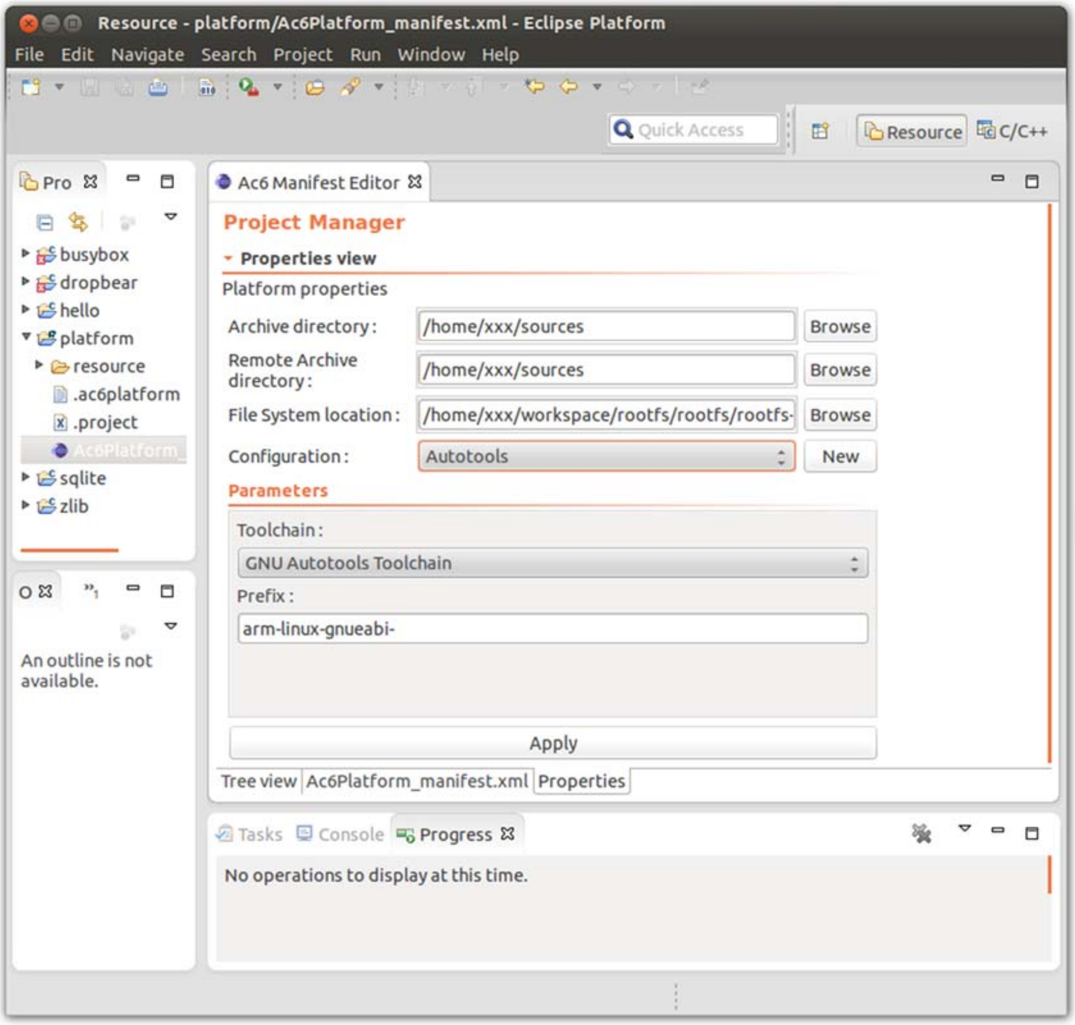
ac6 tools

System Workbench for Linux

Build conveniently your embedded Linux systems

To help you solve the problems of creating full-fledged embedded Linux platforms, Ac6 has developed *System Workbench for Linux*, a full, Eclipse-based, Integrated Development Environment that to create your embedded Linux systems.

System Workbench for Linux helps you develop both your platform and your applications. It provides a convenient, comprehensive and powerful environment for embedded system developers.



Archive directory: /home/xxx/sources

Remote Archive directory: /home/xxx/sources

File System location: /home/xxx/workspace/rootfs/rootfs/rootfs-

Configuration: Autotools

Toolchain: GNU Autotools Toolchain

Prefix: arm-linux-gnueabi-

An outline is not available.

No operations to display at this time.

A Complete environment

System Workbench for Linux offers an integrated graphical environment to provide intuitive use. The range of tools gives the possibility to fully-configure and to build the kernel and the applicative layer to generate a complete Linux distribution according your expectations. All the tools contributes to accelerate, to make the development setting up easier and successful.

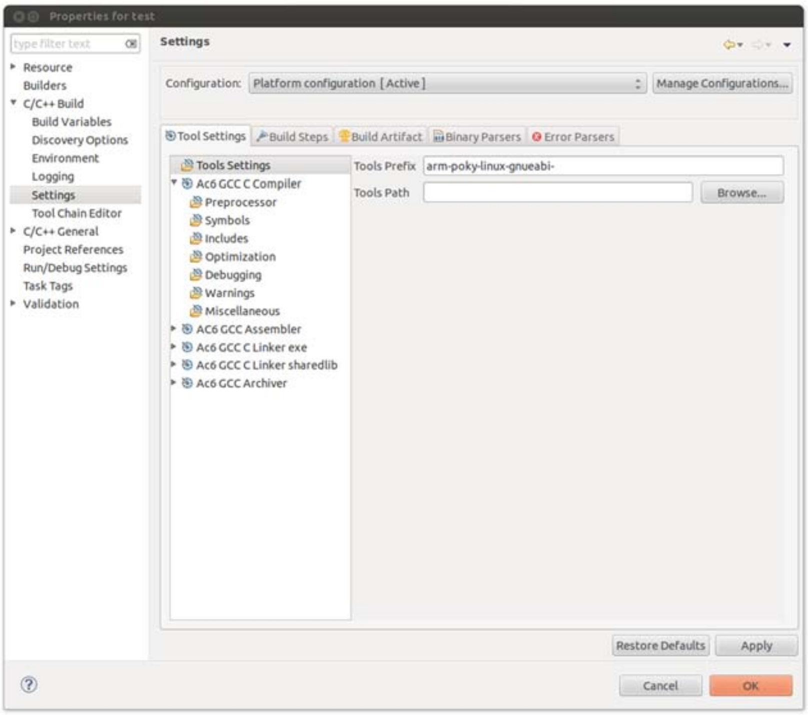
Eclipse-based

System Workbench for Linux is an Eclipse plug-in. Eclipse is an open source integrated development environment used and contributed by many companies. It provides a graphical interface and automatic tasks. This plug-in is working and is fully-integrated with the latest version of Eclipse. Ac6 System Workbench enhances the Eclipse features to support the development for embedded world.

Define your target platform

System Workbench for Linux has been designed to match developers’ expectations : user-friendly interface, sensible default settings and flexible configuration. The parts of the target platform definition consist of :

- Processor Architecture definition
- Cross-compiler toolchain
- Communication media (network, serial line...)
- Debugger configuration (software debugger, JTAG debugger)
- Package configuration settings for userland programs
- Filesystem configuration (support, formats, partitioning...)



Configuration: Platform configuration [Active]

Tools Prefix: arm-poky-linux-gnueabi-

Tools Path:

Prebuilt Toolset

Ac6 provides several prebuilt toolsets which work with your devices. These toolsets are compatible with Eclipse CDT and can be configured through the projects properties.

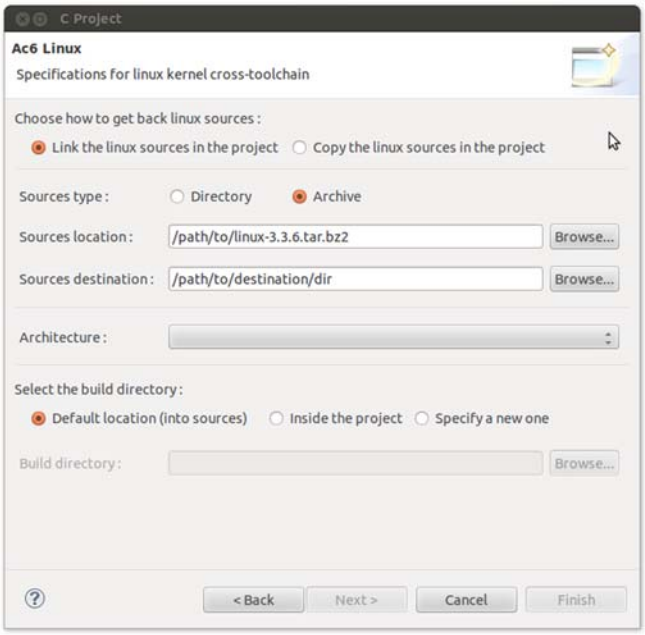
These toolchains are official toolchains, validated by an independent organization, like Linaro for ARM toolsets.

The cross-compiler toolchain can also be defined by using the standard Eclipse CrossGCC toolchain connected to your own GCC-based compiler suite.

Configure, compile and install your Linux kernel

From the provided environment, the steps to configure and to compile your Linux Kernel are simplified. The architecture supported by the kernel can be selected and numerous predefined and board-specific configurations can be set to match to the target device.

Besides getting the Linux Kernel ready, the IDE supports compiling and installing out-of-tree Linux modules (such as specific drivers) according to the kernel configuration. Patches (such as RT_PREEMPT or Xenomai for Real-Time) can also be applied to enhance the Kernel features.



C Project

Ac6 Linux

Specifications for linux kernel cross-toolchain

Choose how to get back linux sources :

● Link the linux sources in the project ○ Copy the linux sources in the project

Sources type : ○ Directory ● Archive

Sources location : /path/to/linux-3.3.6.tar.bz2

Sources destination : /path/to/destination/dir

Architecture :

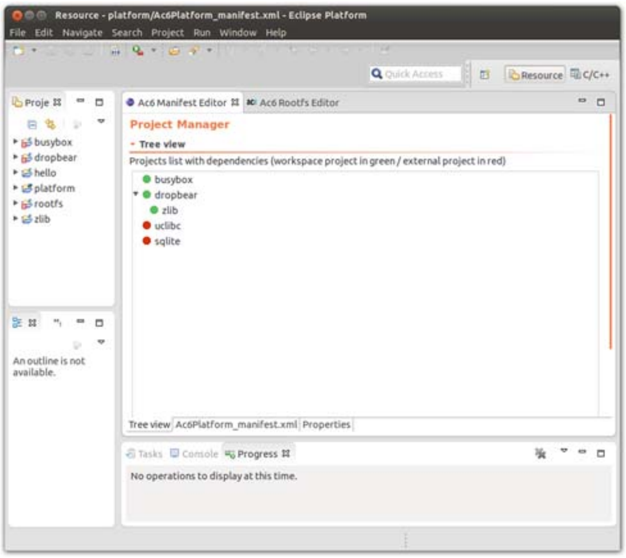
Select the build directory :

● Default location (into sources) ○ Inside the project ○ Specify a new one

Build directory :

Select, Configure and install the user-level package

The workbench offers to customize the user-level packages from a set of predefined configurations (minimal, network appliance ...), to add or remove, configure, build and install packages from your platform via a dedicated editor and project manager. The package analysis is also integrated to avoid package dependency issues and to get a clean and working user-level applicative layer.



Resource - platform/Ac6Platform_manifest.xml - Eclipse Platform

Ac6 Manifest Editor | Ac6 Rootfs Editor

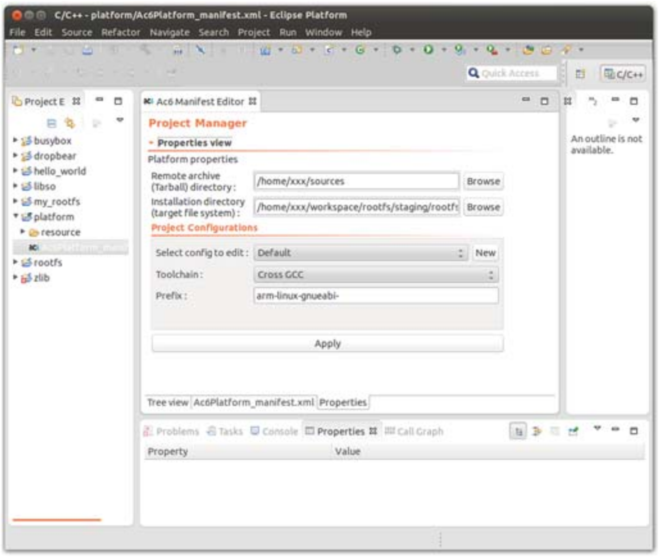
Project Manager

Tree view

Projects list with dependencies (workspace project in green / external project in red)

- 🟩 busybox
- 🟩 dropbear
  - 🟩 zlib
- 🟥 uclibc
- 🟥 sqlite

No operations to display at this time.



C/C++ - platform/Ac6Platform_manifest.xml - Eclipse Platform

Ac6 Manifest Editor

Project Manager

Properties view

Platform properties

Remote archive (Tarball) directory : /home/xxx/sources

Installation directory (target file system) : /home/xxx/workspace/rootfs/staging/rootfs

Project Configurations

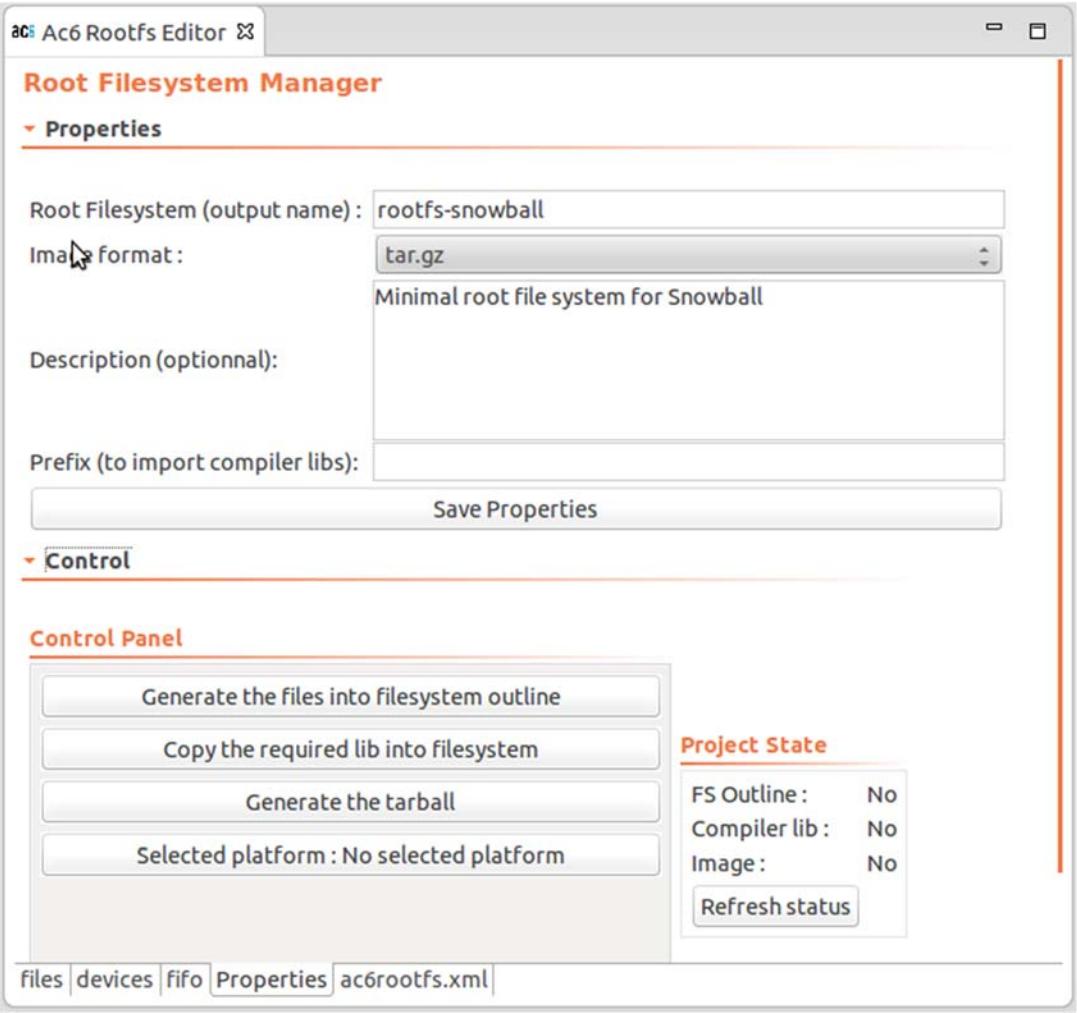
Select config to edit : Default

Toolchain : Cross GCC

Prefix : arm-linux-gnueabi-

Target platform management

To create a platform one needs to have each component well-configured and built. The target platform can be managed in the integrated environment. The workbench guides the user from the bootloader definition to the filesystem images setting up. You can configure and build u-boot for your platform, build automatically all the requested packages. A root filesystem editor is provided to make following steps easier: going through filesystem definition (files, devices, fifo) and installation, finding and copying the required libraries from toolchain thanks to a provided script and generating a ready-to-use filesystem image with the built packages.



Ac6 Rootfs Editor

Root Filesystem Manager

Properties

Root Filesystem (output name) : rootfs-snowball

Image format : tar.gz

Description (optionnal): Minimal root file system for Snowball

Prefix (to import compiler libs):

Control

Control Panel

Project State

FS Outline : No

Compiler lib : No

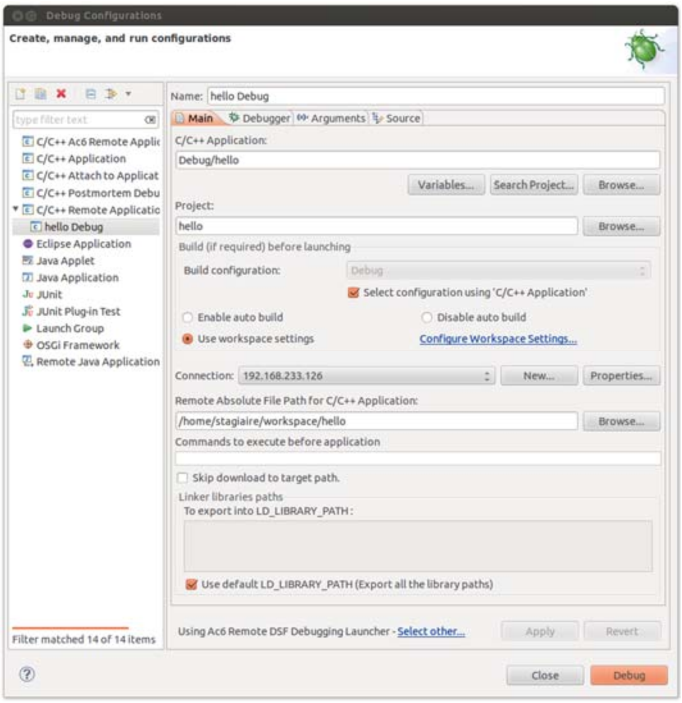
Image : No

Save Properties
Generate the files into filesystem outline
Copy the required lib into filesystem
Generate the tarball
Selected platform : No selected platform
Refresh status
files
devices
fifo
Properties
ac6rootfs.xml

Debug remote application

Ac6 System Workbench includes debugger configuration patterns to provide software local or remote cross-debugging. It enables to download application into the target and using every features of the debugger such as running in single-step, setting up breakpoints, memory or registers examination or monitoring disassembled code. The debug through a JTAG probe is planned.

The advanced debug configuration is predefined according the toolchain chosen. Further parameters can be applied such as remote workspace (working on target) or shared library selection (to exclude the non-needed library).



Debug Configurations

Create, manage, and run configurations

Name: hello Debug

C/C++ Application: Debug/hello

Project: hello

Build (if required) before launching

Build configuration: Debug

☑ Select configuration using 'C/C++ Application'

○ Enable auto build　　○ Disable auto build

● Use workspace settings

Connection: 192.168.233.126

Remote Absolute File Path for C/C++ Application: /home/stagiaire/workspace/hello

Commands to execute before application

□ Skip download to target path.

Linker libraries paths

To export into LD_LIBRARY_PATH :

☑ Use default LD_LIBRARY_PATH (Export all the library paths)

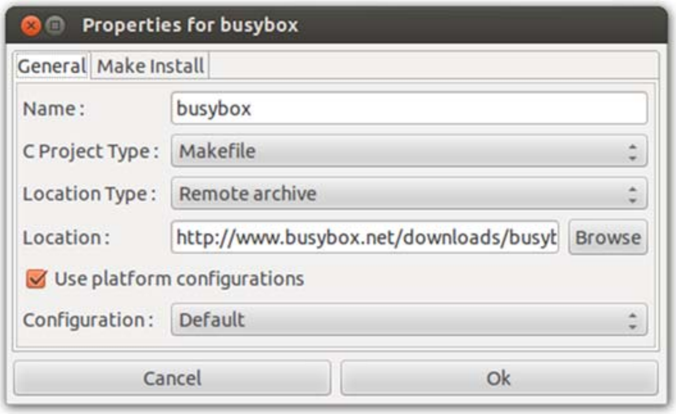
Filter matched 14 of 14 items

Using Ac6 Remote DSF Debugging Launcher - Select other...

Teamwork

Using Ac6 System Workbench allows creating a personalized platform, to manage and to configure the application projects. This platform can also be managed and easily shared between developers. As an Eclipse project, every Eclipse collaboration tools (such as Subversive) are compatible to increase your team productivity.

An application platform management is provided to configure and save the platform information into a manifest file. From this manifest file, the entire application platform can be loaded on another development station. Moreover, the Ac6 System Workbench interface is able to import Eclipse project from SVN, CVS, GIT, archive or remote archive which helps to share the base platform without copying the sources file from all the contents.



Properties for busybox

Name : busybox

C Project Type : Makefile

Location Type : Remote archive

Location : http://www.busybox.net/downloads/busyt

☑ Use platform configurations

Configuration : Default

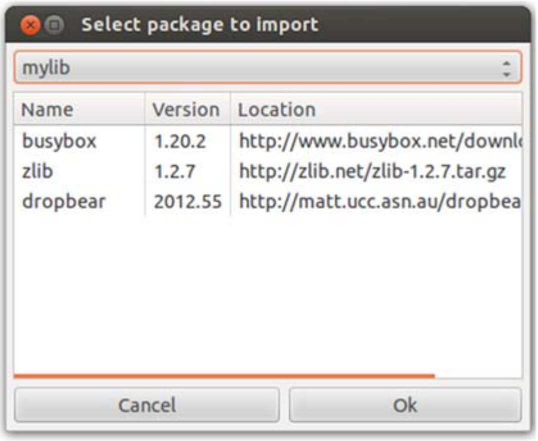
Technical Support

Ac6 System Workbench is a product currently used by Ac6 team. Ac6 provides you active support. The project philosophy is to provide tools that help the developers in to go through time-consuming phase, but at same time, to provide flexibility in configuration. Please contact us to discuss your needs and your suggestions to improve the product.

Libraries Manager

System Workbench for Linux, includes a pack of platform libraries and a libraries editor. A platform library consists of a set of package definitions (name, version, location, dependencies) which can be imported into your embedded platform as packages.

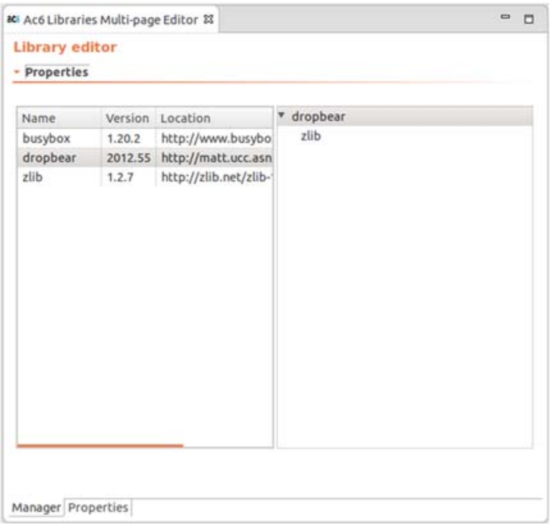
With this feature, the platform development becomes faster and it prevents from mistakes in the user-level packages environment definition. The library selected content is fetched into the platform as a platform element



Select package to import

mylib

Name	Version	Location
busybox	1.20.2	http://www.busybox.net/downl
zlib	1.2.7	http://zlib.net/zlib-1.2.7.tar.gz
dropbear	2012.55	http://matt.ucc.asn.au/dropbea



Name	Version	Location
busybox	1.20.2	http://www.busybo
dropbear	2012.55	http://matt.ucc.asn
zlib	1.2.7	http://zlib.net/zlib-

A libraries editor is also provided to allow you to easily and freely create your own platform library project through the Eclipse interface and to export and share it as a platform library file. The library ensures validity of the package properties, its version and its dependencies.

To help the users to begin their Linux embedded platform development, we provide a set of libraries dedicated to embedded projects on the officially supported targets. It gives a wide list of packages required in small file systems and packages which are often installed.