

System Workbench for Linux Classic Edition

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 – Classic Edition*, 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.

😣 🗐 🗊 Resource - p	latform/Ac6Platform_ma	anifest.xml - Eclipse Platform				
File Edit Navigate	Search Project Run W	indow Help				
📑 🔻 🔛 🕼 🗁 🗎	👬 💁 🔻 🗀 🔗 🔻 💡	최 > 전 > *> 수 > -> -> :알				
		Q Quick Access	e	Resourc	e 🚾 C/	'C++
Pro 🛛 🗖 🗖	Ac6 Manifest Editor 8	3			•	
트 🔄 😜 🔻	Project Manager					- 11
🕨 🔐 busybox	 Properties view 					- 11
dropbear	Platform properties					- 11
▶ 🚰 hello ▼ 😅 platform	Archive directory :	/home/xxx/sources	Browse			
resource	Remote Archive directory :	/home/xxx/sources	Browse			
incopract in acceptation in a copract	File System location :	/home/xxx/workspace/rootfs/rootfs/rootfs-	Browse			
Ac6Platform_	Configuration :	Autotools ‡	New			- 11
Sqlite	Parameters					- 11
P 🗁 ZUD	Toolchain :					- 11
	GNU Autotools Toolo	chain	÷			- 11
o¤ "₁ □ □	Prefix :					- 11
_ ⊽	arm-linux-gnueabi-					- 11
An outline is not available.						
		Apply				- 11
	Tree view Ac6Platform	manifest xml Properties				-'
						\neg
	🖉 Tasks 🖳 Console 🖷	Progress 🛛		*	~ •	
	No operations to displa	ay 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...)

🗵 🗉 Properties for test	t	
type filter text 🛛 🕱	Settings	⇔ ▼ ⇔⊤ ▼
 Resource Builders C/C++ Build Build Variables 	Configuration: Platform configuration [Active]	C Manage Configurations
Discovery Options Environment Logging Settings Tool Chain Editor > C/C++ General Project References Run/Debug Settings Task Tags > Validation	Tool Settings PBuild Steps PBuild Artifact PBuild Artifact PBuild Artifact PBuild Artifact Tools Settings Tools Prefix arm-poky-linux-gnueabi- Tools Prefix arm-poky-linux-gnueabi- <td>Browse</td>	Browse
	R	estore Defaults Apply
?		Cancel

Prebuilt Toolset

Ac6 provides several prebuilt toolsets which work with the devices used in our training sessions. These toolsets are compatible with Eclipse CDT and can be configured through the projects properties.

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 bac	k linux sources : rrces in the project 〇 Copy the linux sources in the project	: D
Sources type :	O Directory	
Sources location :	/path/to/linux-3.3.6.tar.bz2	Browse
Sources destination :	/path/to/destination/dir	Browse
Architecture :		*
Select the build directo Default location 	ory : \bigcirc Inside the project \bigcirc Specify a new on	e
Build directory :		Browse
?	<back next=""> Cancel</back>	Finish

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 - pl	atform/Ac6Platform_manifest.xml - Eclipse Platform	😣 🗐 🕤 C/C++ - platfo	orm/Ac6Platform_manifest.xml - Eclipse Platform	
File Edit Navigate S	earch Project Run Window Help	File Edit Source Ref	actor Navigate Search Project Run Window Help	
1 - E & O E)] 🂁 📲 😂 🖋 📲 및 포함 포함 수 포함 포함 권			* 🙋 😂 🔗 *
	Q Quick Access 🗈 🗈 Resource 🗟 C/C++	<u>9</u> = 9 = 9 + 9 +	Quick	Access
ြာ Proje 🛿 📍 🗖	Ac6 Manifest Editor 22 Ac6 Rootfs Editor	🕒 Project E 🕮 📟 E	aci Ac6 Manifest Editor 🗱	
😑 😫 🔝 🔻	Project Manager	840	Project Manager	÷ ₹
busybox	• Tree view	► 🥵 busybox	 Properties view 	An outline is not available.
dropbear	Projects list with dependencies (workspace project in green / external project in red)	Second and the sec	Platform properties	
 Bello Platform 	busybox	► 😂 libso	Remote archive (Tarball) directory : //home/xxx/sources Browse	
► 🔂 rootfs	• zlib	► Smy_rootfs	Installation directory (target file system): //home/xxx/workspace/rootfs/staging/rootfs Browse	
- 200	• uclibc	► ▷ resource	Project Configurations	
	• sqlite	aci Ac6Platform_mai	Select config to edit : Default 2 New	
		► Pootrs ► 🗟 zlib	Toolchain : Cross GCC ‡	
			Prefix : arm-linux-gnueabi-	
95 M				
			Apply	
An outline is not			- Appy	
available.				
			Tree view Ac6Platform_manifest.xml Properties	
	Tree view Ac6Platform manifest.xml Properties			
			🖹 Problems 🕗 Tasks 🖳 Console 🔲 Properties 🗱 🛲 Call Graph	
	🕢 Tasks 🖳 Console 🖏 Progress 🖾 🦂 🗎 🗖		Property Value	
	No operations to display at this time.			

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.

aci Ac6 Rootfs Editor 🛿			-	
Root Filesystem Manage	r			
 Properties 				
Root Filesystem (output name) :	rootfs-snowball			
Ima a format :	tar.gz		\$	
	Minimal root file system for	Snowball		
Description (optionnal):				
,				
Prefix (to import compiler libs):				
	Save Properties			
- Control				
Control Panel				
Generate the files into	filesystem outline			
Copy the required li	b into filesystem	Project State		
Generate th	e tarball	FS Outline :	No	
Selected platform : No	selected platform	Image :	No	
		Refresh status		
files devices fifo Properties ace	frootfs.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).



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.

😣 🗉 Propertie	es for busybox
General Make In	stall
Name :	busybox
C Project Type :	Makefile ‡
Location Type :	Remote archive ‡
Location :	http://www.busybox.net/downloads/busyt Browse
👿 Use platform	configurations
Configuration :	Default
Ca	ncel Ok

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 – Classic Edition, includes a pack of platform libraries and a libraries editor. A platform library consists on a set of package definitions (name, version, location, dependencies) which can be imported into your applicative 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

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

Mamo			
Name	Version	Location	▼ dropbear
busybox	1.20.2	http://www.busybo	zlib
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 applicative platform development, we provide a set of libraries dedicated to embedded projects. It gives an exhaustive list of the packages required in minimal file system and packages which are often installed.