

Introduction to Linux-based solution for embedded software development

Section 1 Eddy Real-Time Linux, Lemonix

Section 2 Eddy Integrated Development Environment, LemonIDE

Section 3 Eddy Utility Programs

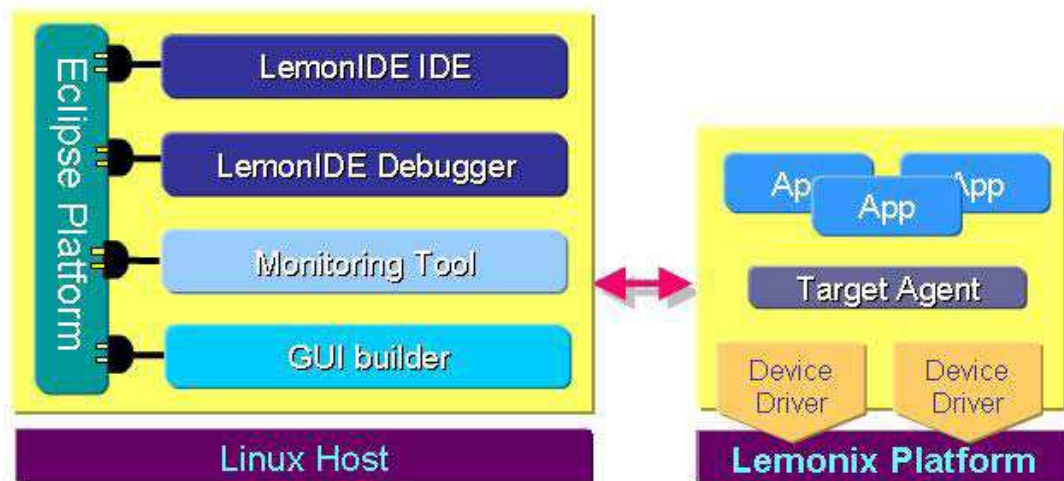
Eddy Integrated Development Environment, LemonIDE™ for Embedded Software System Development

Software development in embedded systems is usually done on widely used computer systems such as desktop PC due to embedded system's lack of resources and performance. Such development environment is called Cross-Development Environment where the computer system in which the development takes place is called the host system and the embedded system in which the developed software is executed is called the target system. The host and target system allows users to download and execute, debug or monitor the interactively developed program in the target system through communication devices such as Ethernet or serial.

In this article, we would like to introduce the LemonIDE™, an Eclipse based integrated development environment with GUI environment for Lemonix™.

1. Summary

LemonIDE™ is an Eclipse-based integrated development environment for embedded software that provides GUI (Graphic User Interface) environment in which applications and firmware for real-time Linux system, Lemonix, can be easily developed. It also resolves former problems in TUI based development environment by providing an environment which enables all developing processes such as GNU C/C++ compiler, source code editor, debugger and etc. to be executed on a single program. This means that users don't have to be familiar with each compiler or debugger's program commands. Not only does LemonIDE™ provide the convenience to execute successive processes from compiling to debugging with only mouse clicks by providing a unified GUI but it also provides almost every development abilities such as automatic Makefile creation, automatic source completion, remote debugging, plug-in support, target system monitoring and etc. Furthermore, it is provided in both Linux and Windows version so that developers can choose whichever is more convenient for them.



[* LemonIDE™ Structure]

Therefore, users can quickly and easily develop embedded software such as firmware and other application programs that runs in Lemonix™ based embedded systems (target systems) using LemonIDE™ in Linux or Windows host system.

Lemon IDE™ contains following tools.

* **LemonIDE™ IDE :**

Designed to support project-based C/C++ source program editing, cross-compiling, remote execution, connection with GUI builder.

* **LemonIDE™ Debugger :**

Debugs programs developed with LemonIDE™. Supports stop-point based debugging and trace point based debugging which does not need stopping the program.

* **Monitoring tools :**

Shows information on target system's memory, process, resource, battery usage and etc.

* **Target Agent :**

Executed on target system. Provides functions such as file transmitting or remote execution in connection with host system.

2. Functions

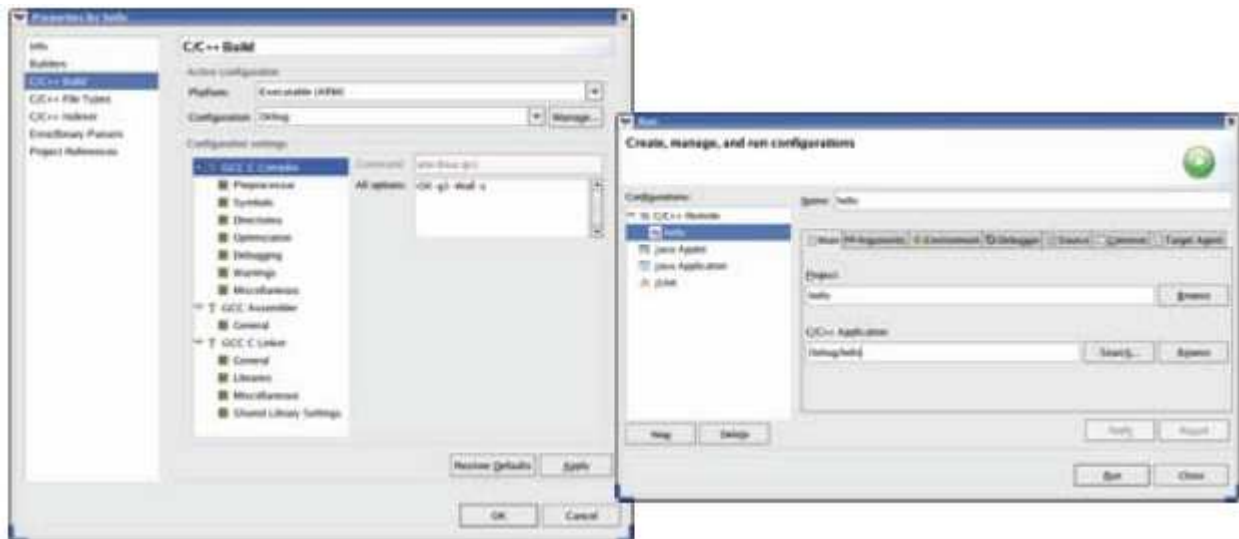
Primary function of each tool contained in LemonIDE™ is as follows.

* **IDE**

IDE provides source editing, cross-building, remote execution of project based C/C++ application programs and export, import of application programs. Eclipse platform contains text editor and project manager. Text editor is an editor for general text files, not for a specific program source and project manager also only has abilities to handle general projects, not specific projects for application program development. Lemonix™ means a project manager and text editor specialized in C/C++ language which is widely used for embedded

software development. The project manager and source editor designed for remote development performs its task of remote application program development by extending project manager and text editor contained in Eclipse platform.

Cross-compile toolchain is a packaged tool based on open sources such as GNU's binutils, glibc or gcc designed to support target system's CPU. Cross-compile toolchain contains compiler that can compile and link C/C++ source programs and binary utility that can control the compiled binary codes. IDE project manager used the toolchain to compile completed source programs.



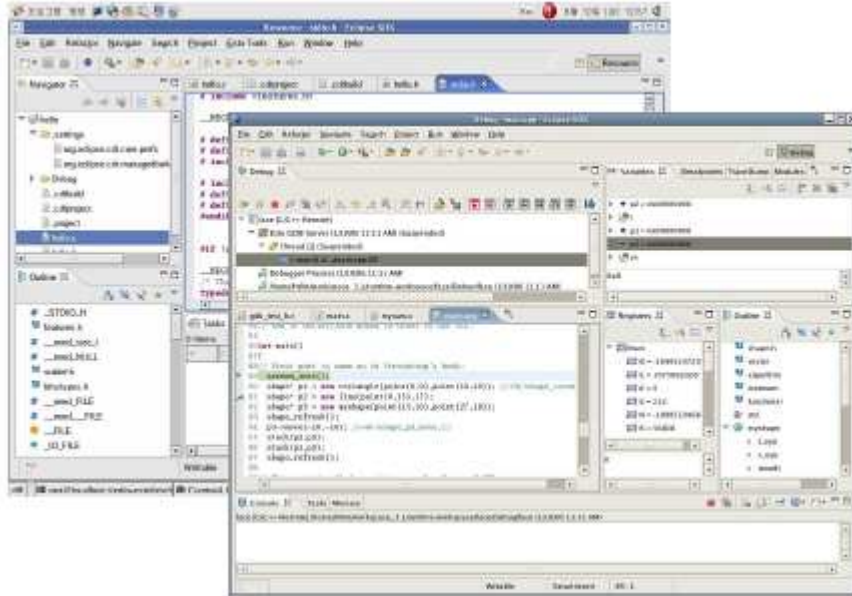
[* Project Build & Remote Execution Configuration]

* Debugger

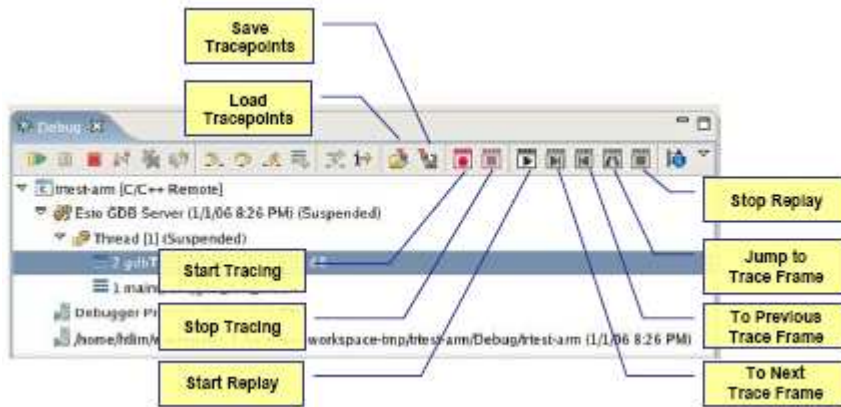
Remote Debugger allows user to debug programs running in target boards in host systems. For remote debugging, the program developed in the host system needs to be sent to the target system, executed with the target's debug agents (gdbserver) and the host's debugging engine and target's debug agents need to be connected. However, it is inefficient to repeat these steps on every debugging. LemonIDE™'s remote debugger automatically goes through above steps and remote debugging can be easily done as local debugging.

Furthermore, the remote debugger provides non-stop debugging capability. It is useful in cases in which a time-sensitive program does not show errors or time out occurs and debugging gets complicating when it is debugged by stopping the program to a stop point. By using trace points, program status can be examined after the program is executed without stopping the program. Add the target such as a variable or register to the trace point and execute the program. Whenever the program meets the trace point, it will stop and check the value of the variable or the register that user added, record it and continue. After the tracing is over, users can examine the traced values just as if they are debugging. LemonIDE™ remote Debugger supports the non-stop debugging capability that is embodied in GNU GDB. As non-stop debugging capability was formerly applied only to debug engine but not in debug agent, we have applied the non-stop debugging capability to the debug agent.

JTAG based debugger is used to debug the hardware of an embedded system without any special devices and with low cost. It enables debugging by activating the JTAG capability in the embedded system using only USB ports embodied in the develop host and provides a practical and easily extendable development environment by plugging-in to Eclipse, an open software integrated development environment.



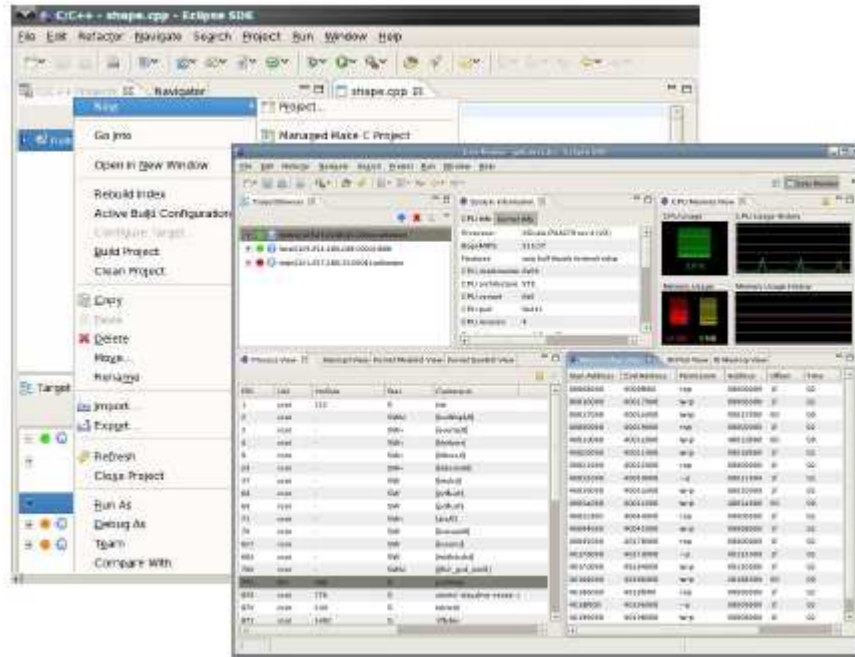
[* Resource View and Visual Source Code Debugger]



[* Debugger base on Start and Stop]

*** Monitoring Tool**

Target Monitoring Tool allows users to monitor various statuses of the target systems in real time. While conveniently managing targets, users can monitor each target's status information or execute, debug developing programs on a target systems. In addition, users can track down what system calls or library functions are being called when developing program is executed on target system.



[* Project Manager and Target Monitor Tool]

* Target Agent

Package setup, build, install, deploy in target systems is done through target setup tool. Target Agent can export the application program that are made with the IDE in the form of a package that can be handled in target setup tool or import it to the IDE so that it can be used to modify the package sources already made.

Provider	SystemBase	'D' Company	'L' Company	'M' Company
IDE	LemonIDE	'E' IDE	'E' IDE	None
Basic Framework	Eclipse	Eclipse	Command line	Command line
Project Management	O	O	—	X
Compiler	GNU C/C++	GNU C/C++	—	uClibc
Makefile Auto Creation	O	O	X	X
Syntax Highlighting	O	O	X	X
Code Auto Completion	O	O	X	X
Source Browsing	O	O	X	X
Remote Control	O	X	X	X
Remote Build	X	X	X	X
App Auto Execution	O	O	X	X
Debugger	gdb	gdb	—	X
Multi-Thread Debugging	O	O	—	X
Multi-core Debugging	X	X	X	X
JTAG Device Support	X	X	X	X

Remote Debugging	O	O	—	X
Target System Monitoring	O	O	—	X
Unit Performance Test	X	O	—	X
Tool Chain	O	O	O	O
Kernel customize	X	X	O	X
File System customize	X	X	—	X
Remote Shell, Telnet	O	O	—	X
plug-in support	O	O	—	X
Host OS	Linux Windows	Window Linux	Linux	Linux
Target OS	Lemonix RTOS	NET+OS	Evolution OS	ucLinux

[* IDE (Integrated Development Environment) Primary Function Comparison Chart]

3. Specifications

LemonIDE™ Specifications are as follows.

* Eclipse

- Eclipse SDK 3.2.2
- CDT core 3.1.2
- CVS core 3.2.2

* Compiler

- GNU C/C++ 3.4.3
- Arm-linux cross compiler

* Editor

- C/C++ source code editor
- C/C++ automatic code completion
- phrase highlighting
- Automatic Makefile creation
- Source web browser
- Multi-file editing with tab interface
- File search and advanced search support
- External editor selection depending on file types
- Precise indication of error statements
- Auto Build Ability
- Remote Execution

*** Debugger**

- GNU gdb 6.3.90
- Visual source code debugger
- Stop point & trace point based debug support
- Remote debugging support
- Multi-thread debugging support

*** Target Monitor Tool**

- Targetview Plug-in 1.0.0
- Monitor Plug-in 1.3.4

*** Host System**

- Linux / Windows installed computer
- JRE 5.0 or higher version required

*** Verified Windows Distro**

- Windows XP SP2
- Windows 2000
- Windows 2003

*** Verified Linux Distro**

- Fedora Core 4, 5, 6
- Redhat 9.0
- Ubuntu Linux 6.x, 7.x
- Debian 4.0
- SUSE Linux Enterprise Server 10.2
- CentOS 4.5
- Asianux

*** Target System**

- Eddy-CPU V2.0
- Eddy-S1/PIN V2.0
- Eddy-S1/DB9 V2.0
- Eddy-S1/DB9-PoE V2.0
- Eddy-S2M/PIN V2.0