Embest IDE for ARM

1.1 About Embest IDE

Embest IDE is an Integrated Development Environment for software cross-development. The Embest IDE is an application that provides a simple and versatile graphical user interface and tools for developing embedded software. It is an Integrated Development Environment (IDE) that facilitates managing and building projects, establishing and managing host-target communication, running and debugging applications. It provides an efficient way for developing embedded applications. Embest IDE comprises the following elements (Figure 1-1):

• An easy-to-used GUI interface

• An integrated source-code editor.

• A project management facility.

• Integrated C and ASM compilers and linker.

• a source-level debugger.

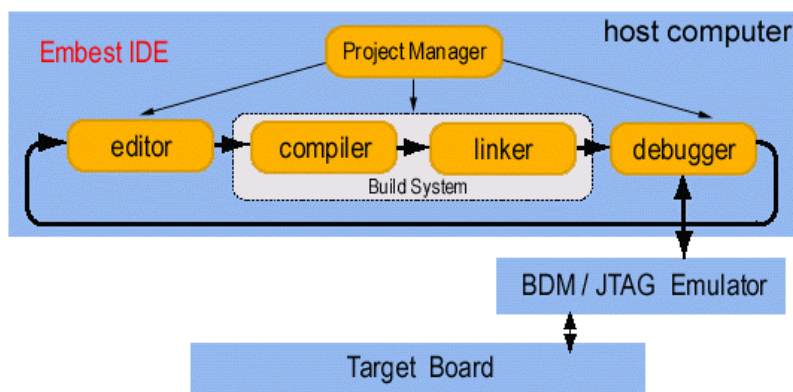• an integrated development environment.

• an ARM simulator

Figure 1-1: Embest IDE Tools

Embest IDE uses host-target cross model for developing embedded applications. Embest IDE runs on the host computer which connectes the target board through debug device (JTAG Emulator). You can edit and build your projects on the host, and create a target executable file with Embest IDE. Download the target file to the target, and then use Embest IDE debugger to debug it through the communication of

debug device connected between the host computer and the target board. Figure1-2 illustrates the host-target cross model with Embest IDE.

When you use Embest IDE for ARM, a debug device——Embest JTAG Emulator or PowerICE is provided.
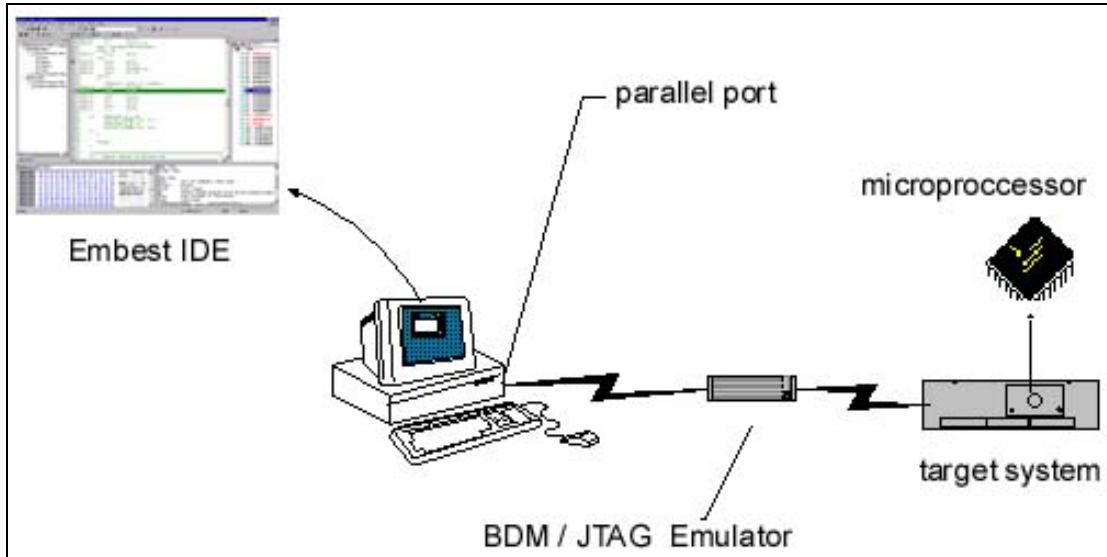

Figure 1-2 the Host-target Cross Model with Embest IDE

1.2 Major Features of Embest IDE

Embest IDE runs under Windows98 and Windows NT 4.0 above. Embest IDE supports target microprocessors debugging on the ARM-core microprocessor families. Embest IDE provides a simple, versatile and graphical user interface for managing your software development projects. You can use Embest IDE for ARM to develop C and ARM assembly language code targeted at ARM and Thumb processors. It speeds up your development cycle by providing:

• Comprehensive project management capabilities

• Code navigation routines to help you locate routines quickly.

Embest IDE enables you to organize source code files, library files, other files, and configuration settings into a project. Each project enables you to create and manage multiple configurations of build target settings.

Major features of Embest IDE:

• Support development language: ANSI C, ARM assembly language.

• User interface: consists of an integrated set of windows, tools, menus, toolbars, directories, and other elements that allow you to create, test, and refine your application. It is just like Microsoft Visual Studio's user interface.

• Source code Editor: Standard text manipulation capabilities; C and ASM syntax-element color highlight; Debugger integration, the editor window tracks code execution; Compiler integration, the project management utility links compiler warnings and errors directly to the affected source in the editor window. Support file print; Capable of search and replace, and performs batch searches in multiple files.

• Project Management: The Embest IDE project facility simplifies organizing, configuring, and building embedded applications. It includes graphical configuration of the build environment (including compiler flags).
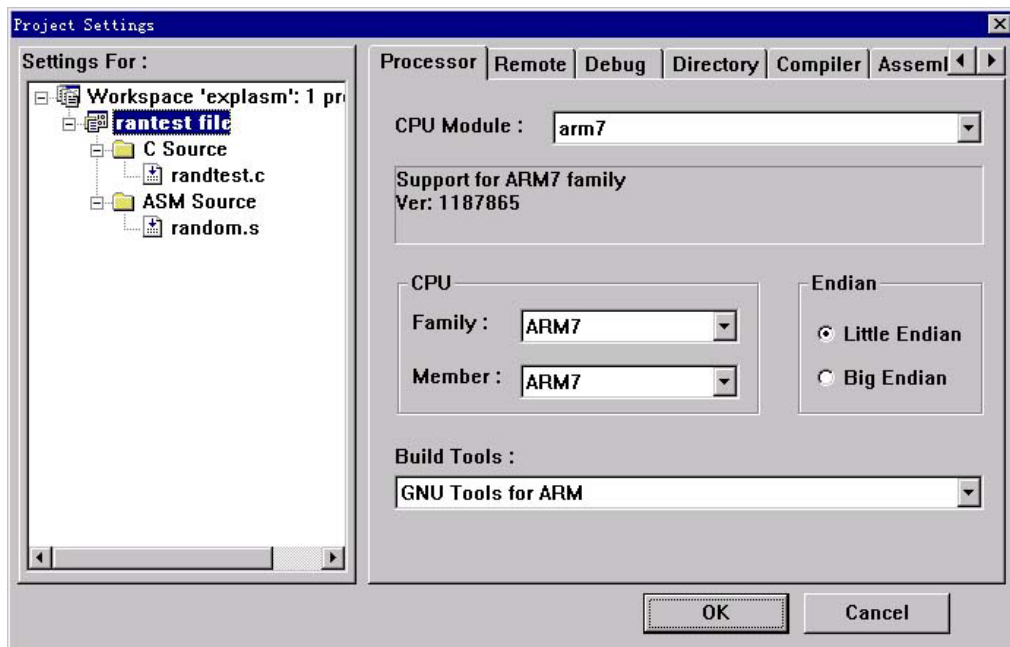

figure 1-3 Project settings dialog box

• Compiler: includes the GNU compiler, as well as a collection of supporting tools that provide a complete development tool chain: cpp, C preprocessor; gcc, C compiler; make, program building automation tool; ld, programmable static linker; as, portable assembler; binary utilities. Embest IDE supports commercial versions of the leading-edge GNU tools originally developed by the Free Software Foundation (FSF). Users of the GNU tools benefit from the innovative FSF development environment as well as from testing and support by Embest Info&Tech Co., LTD. Among other features, the Embest IDE project facility provides a GUI for the GNU tools that is powerful and easy to use.

• Embest IDE debugger: a powerful graphical debugger that enables program loading, executing, running control, and monitoring; a source-level debugger, view your application code as C, as

assembly-level code, or in a mixed mode that shows both; full-featured debugging, provides an exhaustive set of debugging features, designed to make it easy to find and fix bugs; set a breakpoint or clear a breakpoint by single click, supports conditional and command breakpoints; single stepping, "step into" traces execution of every individual instruction even when functions are called, "step over" does not trace into the called function, "step out " brings execution back to the calling function; supports register and variables value display and modify, supports function stack display and memory display, several specialized windows display these debugging information. Figure 7-1 shows these debugging information display windows, when you are debugging, you can access these windows using the "View" menu; Embest IDE supports graphical debug and command-line debug. For complex or unpredictable debugging needs, the command-line interface gives you full access to a wealth of specialized debugging commands.
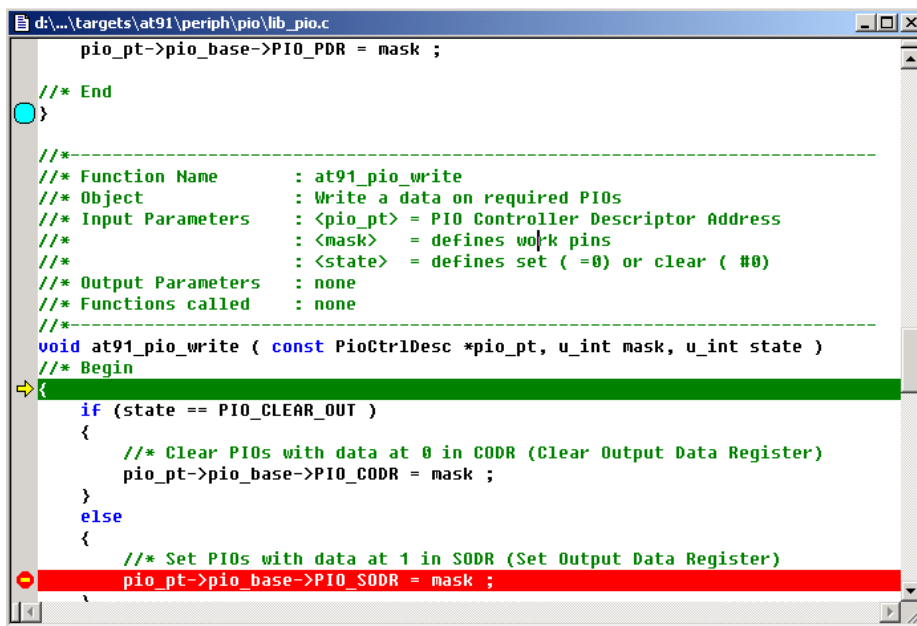
```
d:\...\targets\at91\periph\pio\lib_pio.c                              _ | □ | ×
        pio_pt->pio_base->PIO_PDR = mask ;

  //* End
  }

  //*----------------------------------------------------------------
  //* Function Name     : at91_pio_write
  //* Object            : Write a data on required PIOs
  //* Input Parameters  : <pio_pt> = PIO Controller Descriptor Address
  //*                   : <mask>   = defines work pins
  //*                   : <state>  = defines set ( =0) or clear ( #0)
  //* Output Parameters : none
  //* Functions called  : none
  //*----------------------------------------------------------------
  void at91_pio_write ( const PioCtrlDesc *pio_pt, u_int mask, u_int state )
  //* Begin
⇨ {
      if (state == PIO_CLEAR_OUT )
      {
          //* Clear PIOs with data at 0 in CODR (Clear Output Data Register)
          pio_pt->pio_base->PIO_CODR = mask ;
      }
      else
      {
          //* Set PIOs with data at 1 in SODR (Set Output Data Register)
⊖         pio_pt->pio_base->PIO_SODR = mask ;
```

figure 1-4 debugging show

1.3 User Interface of Embest IDE

The user interface is the portion of the environment where display information and specify action. These topics describe user interface's basic structure of Embest IDE.

To step into Embest IDE, just run Embest IDE.exe. Embest IDE user interface consists of an integrated set of windows, tools, menus, toolbars, directories, and other elements that allow you to create, test, and refine your application. Figure 1-5 shows the main GUI of Embest IDE. The user interface uses standard Windows interface functionality along with a few additional features to make your development environment easy to use. The basic features that you use most often are windows and document views, toolbars, menus, directories, and keyboard shortcuts.

Title column displays current opening file name. Menus column and tool bar column are below title column. The menu bar is a special toolbar at the top of the screen that contains menus such as File, Edit, and Build. The standard tool bar appears just below the menu bar. You can move the toolbars to different locations to suit your needs. Workspace window shows file information about current opening projects. Source code window is the usual used window witch you edit and browse your code in. You can open several source code file windows at one time. Output window displays build information, debug information, file search output information and command-line debug input and output. Status column displays detail information about menus and tool bars, it also displays the current line number and column number of the cursor in source code window.
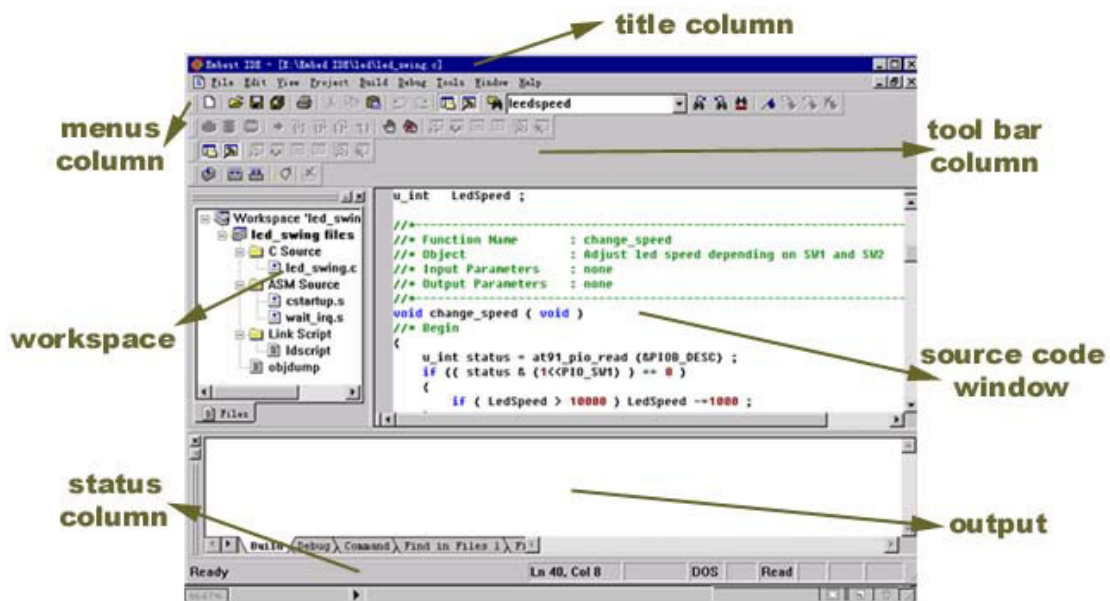
Figure 1-5   Embest IDE Main GUI

Embest Info&Tech Co.,LTD.

Room 509, Luohu Science&Technology Building,
#85 Taining Rd., Shenzhen, Guangdong, China 518020
Tel: +86-755-25635656
Fax: +86-755-25616057
Email: market@embedinfo.com
http://www.embedinfo.com