

***C/C++ Development Kit  
Professional  
for  
TERN 16/32-bit embedded controllers  
with  
Paradigm C++ Professional - TERN Edition***

***Technical Manual***

© TERN, Inc. All Rights Reserved.  
Portions © Paradigm Systems. All Rights Reserved.



1950 5<sup>th</sup> Street, Davis, CA 95616, USA  
Tel: 530-758-0180 Fax: 530-758-0181  
**Internet Email: [sales@tern.com](mailto:sales@tern.com)**

**<http://www.tern.com>**

## COPYRIGHT

A-Engine™, A-Core™, V25-Engine™ are trademarks of TERN, Inc.  
Paradigm C++ Professional - TERN Edition™ are trademarks of Paradigm  
Systems Inc. and TERN, Inc.

Version 1.10

December 5, 2012

No part of this document may be copied or reproduced in any form or by any means  
without the prior written consent of TERN, Inc.



© 2000

1950 5<sup>th</sup> Street, Davis, CA 95616, USA

Tel: 530 758 0180 Fax: 530 758 0181

**Internet Email: [sales@tern.com](mailto:sales@tern.com)**

**<http://www.tern.com>**

## Important Notice

**TERN** is developing complex high technology integration systems. These systems are integrated with software and hardware which are not 100% defect free. **TERN products are not designed, intended, authorized, or warranted to be suitable for use in life-support applications, devices or systems or other critical applications.** **TERN** and the Buyer agree that **TERN** will not be liable for incidental or consequential damages arising from the use of **TERN** products. It is the Buyer's responsibility to protect life and property against incidental failure.

**TERN** reserves the right to make changes and improvements to its products without providing notice.

**TERN** will be responsible for technical support of this kit, NOT Paradigm. Do not contact Paradigm for technical support assistance.

# Chapter 1: Introduction

## 1.1 Description

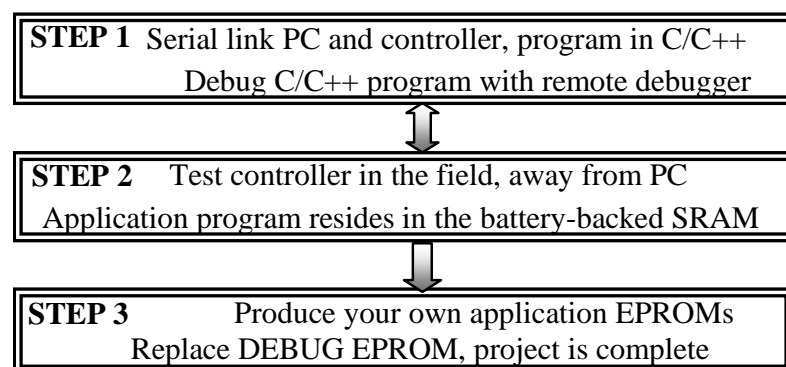
Welcome to TERN C/C++ Evaluation (EV) or Development Kit (DV) for 16/32-bit embedded controllers. TERN controllers are based on PC compatible processors, easy to program in C/C++. The use of ANSI C/C++ compilers for your projects will support your development needs now and into the future.

TERN's EV/DV Kit supports a complete user-friendly integrated development environment for embedded application. It integrates Paradigm C++ Professional - TERN Edition, a state-of-the-art integrated development environment (IDE), including a C/C++ compiler, remote debugger, C libraries, I/O drivers, configuration files, project files, sample programs, technical manuals, power supply, serial cable, remote debug kernel, and a TERN controller.

Within the IDE, you can create, debug, and deploy real-time embedded system applications without resorting to the use of external tools. If you are used to running separate editor, debugger, make, and other tools to get a job done, then you are in for a real treat with Paradigm C++ Professional - TERN Edition. After installing the software and serial link connecting your PC with the controller, you can run a sample program "led.c" by simply click the "run" button. The kit takes care of the whole operation steps including compile, link, locate, download, and ready to run. You can single step, set breakpoint, or run your program on the target controller. After debug and setting a jumper on the board, you can take the controller to a standalone field test. After the standalone field test, the DV kit can generate a HEX/BIN file for your EPROM/Flash. It will dramatically reduce your development time and tool investments. You'll be able to put a dedicated system together in record time.

TERN EV/DV kits support a complete product line of 16/32-bit embedded controllers based on AM188/186/586, i386EX, and V25 processors, featuring operational amplifiers, ADC, DAC, real time clock, EE, solenoid drivers, power relays, UART, opto-coupler, LCD, keypads, PCMCIA, Flash, 9-bit network, and quadrature decoders. MemCard-A™ can support up to 1 GB external memory space. With ACTF™ support, you can download a program to a FLASH without burning a ROM.

## 1.2 Development of application software consists of just three easy steps



### STEP 1

You can program TERN controllers from your PC via serial link with an RS232 interface. Your C/C++ program can be remotely debugged over the serial link at up to 115,000-baud rate. You can single-step, set

breakpoint, or run your program. Figure 1.1 illustrates how the A-Engine, VE232, serial cable and PC are connected.

### STEP 2

After debugging your program, changing a single jumper will permit you to test-run the controller in the field, away from the PC, with the application program residing in battery-backed SRAM. The battery should last 3-5 years under normal conditions.

### STEP 3

When the STEP 2 field test is complete, an application HEX or BIN file can be easily generated on your PC. You may produce your own ROM, or download the application file into FLASH.

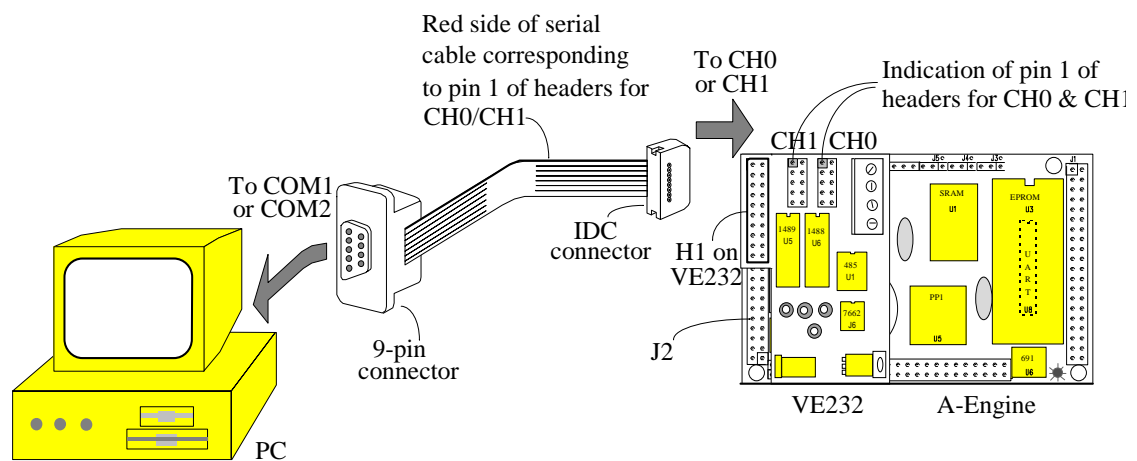


Figure 1.1 Hardware Installation of A-Engine™, VE232™, Serial Cable, and a PC

# Chapter 2: Installation

## 2.1 Software Installation

Installation instructions launch automatically from the Paradigm C++ Professional - TERN Edition CD.

- If you are using V25 based controller, your working directory is C:\TERN\V25
- If you are using Am186/188 based controller, your working directory is C:\TERN\186
- If you are using i386EX based controller, your working directory is C:\TERN\386
- If you are using SC520 (Am586) based controller, your working directory is C:\TERN\586

## 2.2 Hardware Installation

Use A-Engine as an example. Install the VE232™ interface with the H1(10x2) socket connector on the upper half of the J2 (dual row header) of the A-Engine™. Figure 2.1a and 2.1b shows the VE232™ and the A-Engine™ before and after installation. A serial cable (PC-V25) should be used to link the A-Engine™ with your PC. Install the 5x2 IDC connector to the V232™ CH0 header (J2). *Note that the red side of the cable must point to pin 1 of the VE232 J2 header.* A small circle is drawn for indication in the Fig. 2.2. The DB9 connector should be connected to the PC's COM1/COM2. Connect a wall transformer +9V DC output to the VE232™ DC power jack. The LED should blink twice after the A-Engine™ is powered off/on or reset (Figure 2.3).

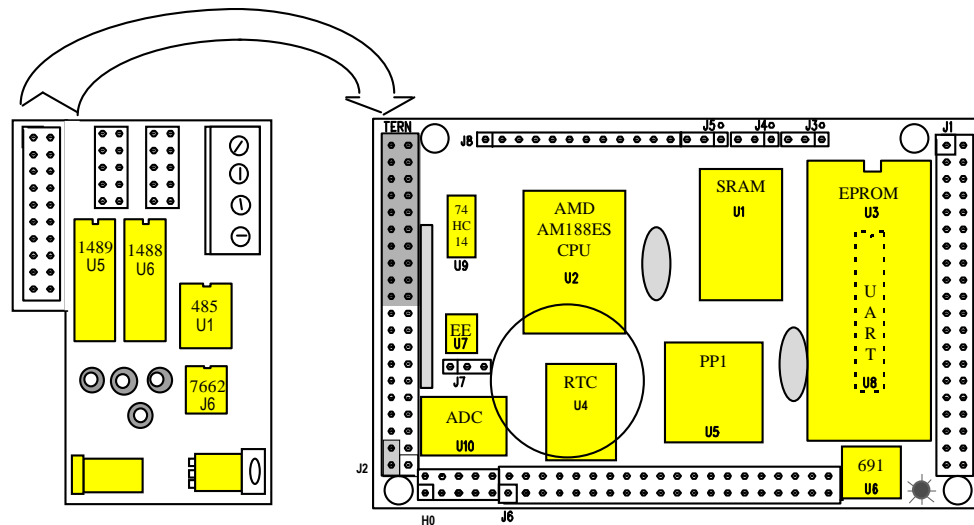


Figure 2.1a. Before installing the VE232™ on the A-Engine™

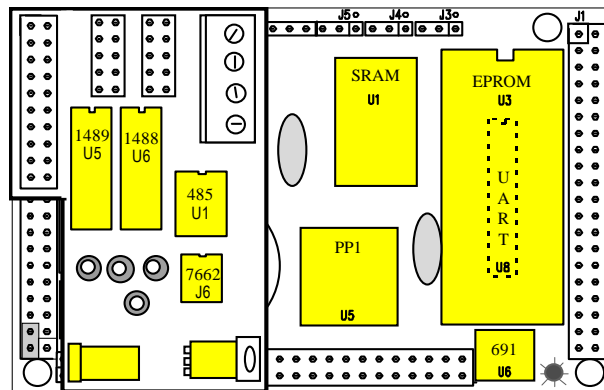


Fig. 2.1b. After installing the VE232™ on the A-Engine™

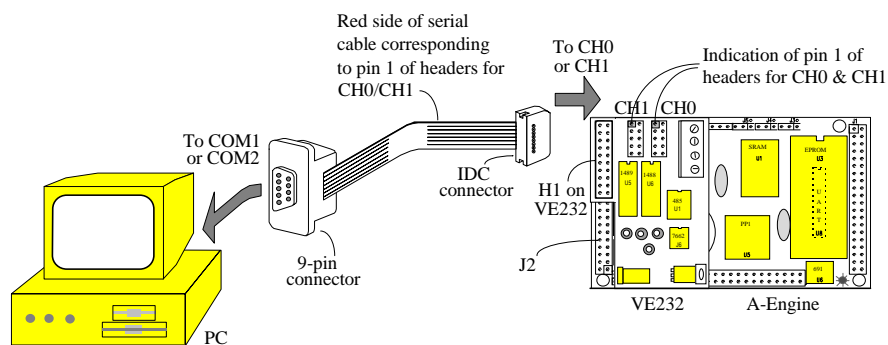


Fig. 2.2. Serial connection between the A-Engine™ and the PC

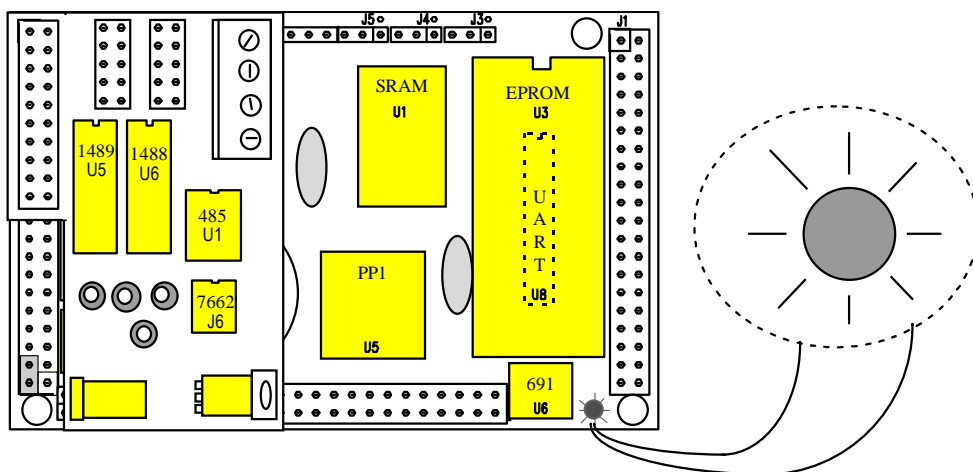


Fig. 2.3. The LED blinks twice indication hardware ready.

## 2.3 USB Adaptor

The EV/DV Kit will come with TERN's USB Adaptor module and USB cable. This module allows serial communication over a USB port. The USB Adaptor is useful for computers that do not support standard serial COM ports. Please refer to the USB Adaptor manual on the Evaluation/Development Kit CD.

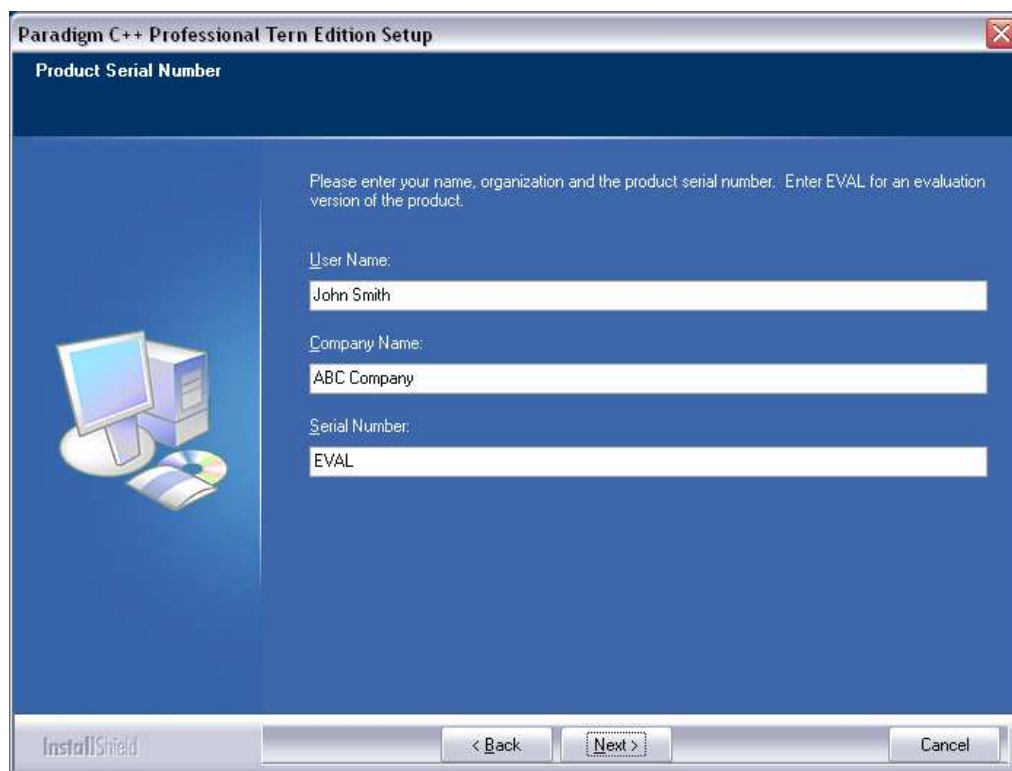


**Fig. 2.3. TERN USB Adaptor**

# Quick Start Guide

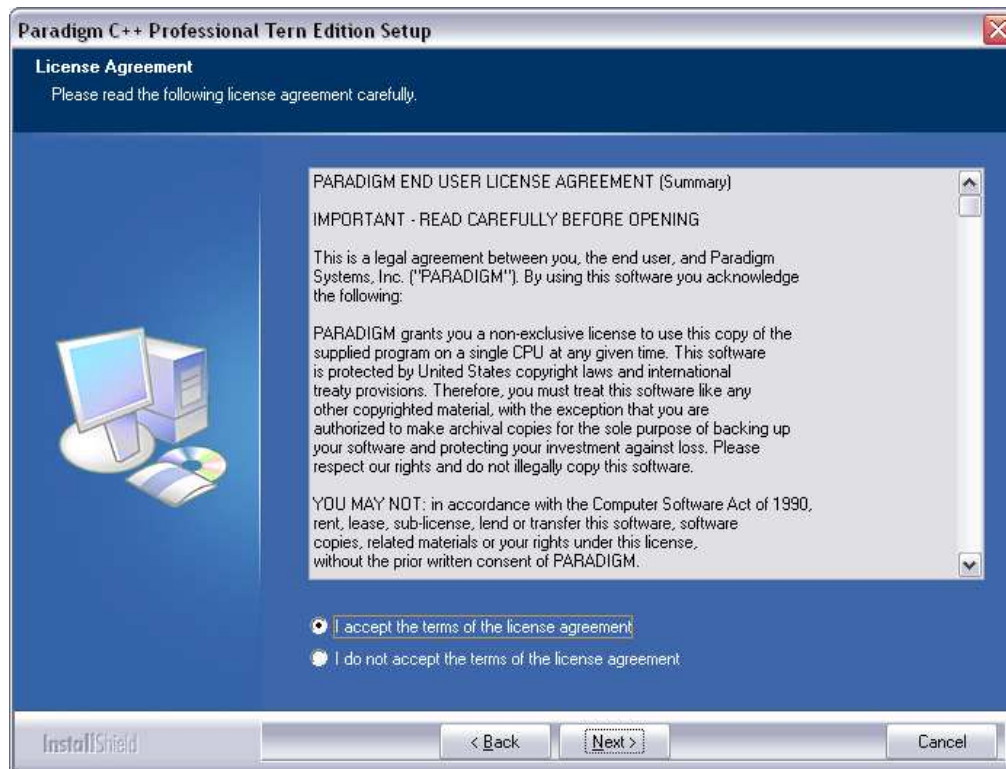
## *How do I install Paradigm C++ Professional – TERN Edition?*

Installation will launch automatically from the TERN CD. If the CD does not run automatically, run Setup.exe from the TERN CD. The setup program will prompt you as shown below.

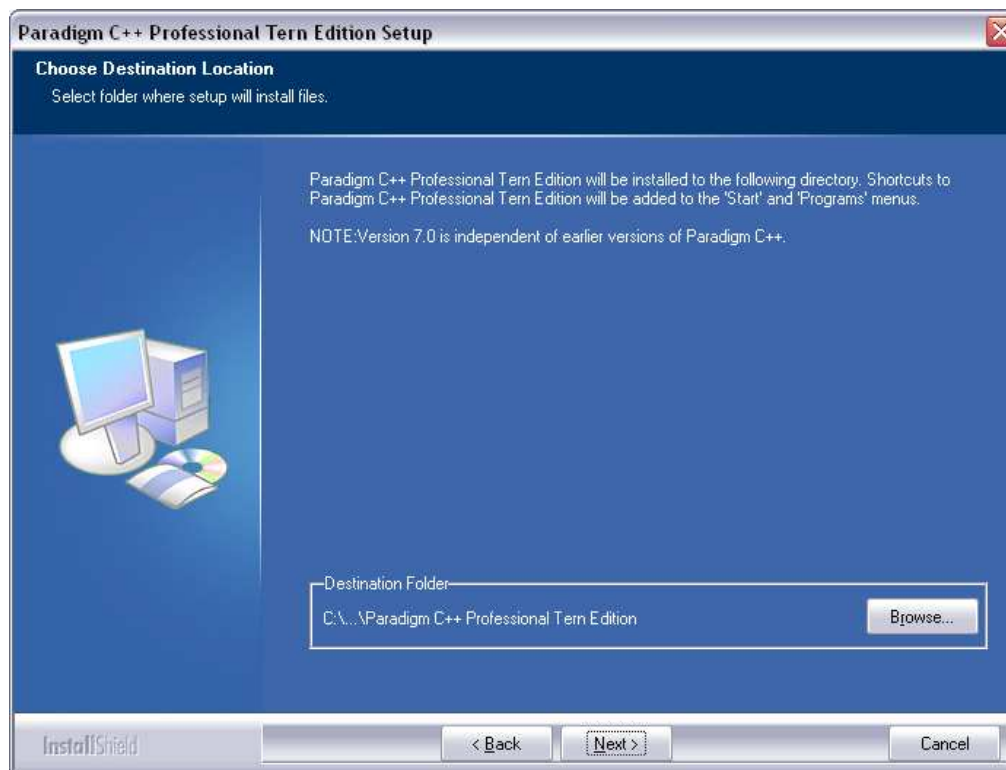


If you purchased a full version DV-P kit from TERN, you will have a serial number from TERN with your order. Enter the serial number to install the full version of Paradigm C++ Professional – TERN Edition. If you do not have a serial number, enter “**EVAL**” to install the Evaluation version. Click ‘Next’ to continue.

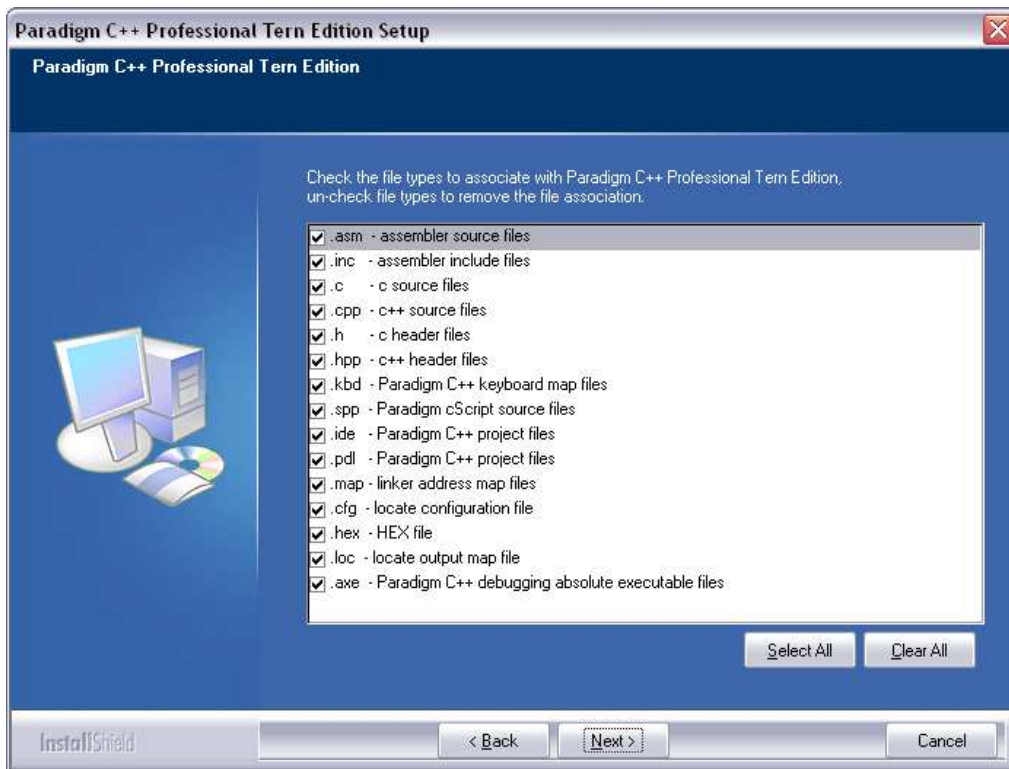




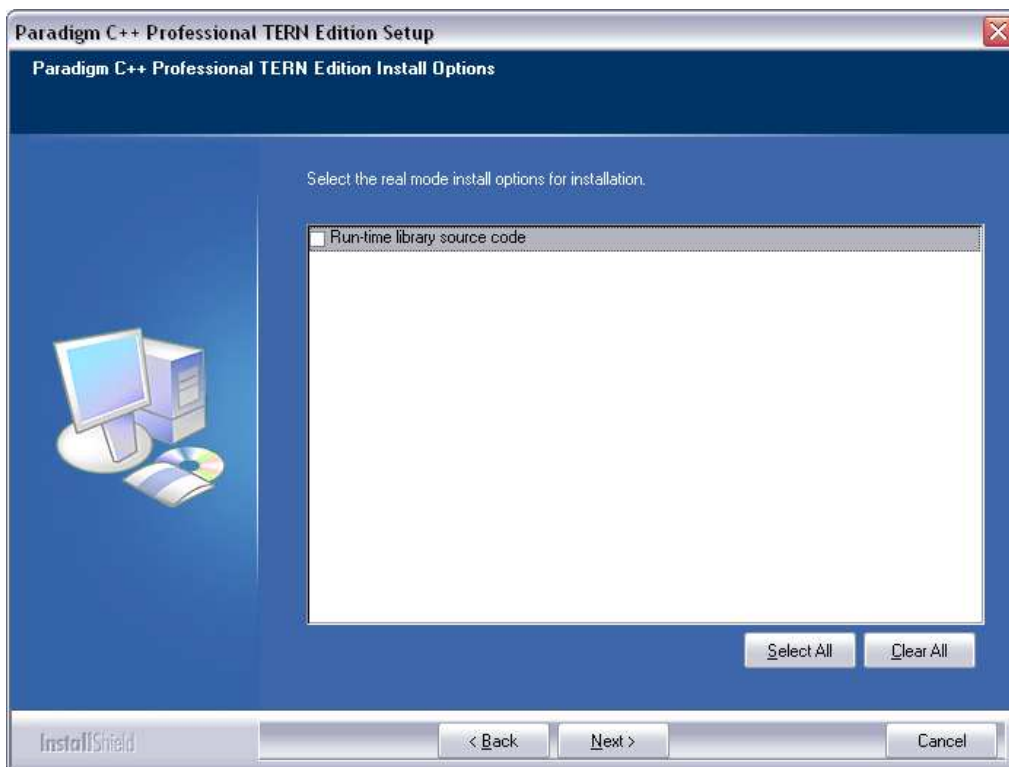
If you accept the terms of the license agreement, click the radio button then click 'Next'.



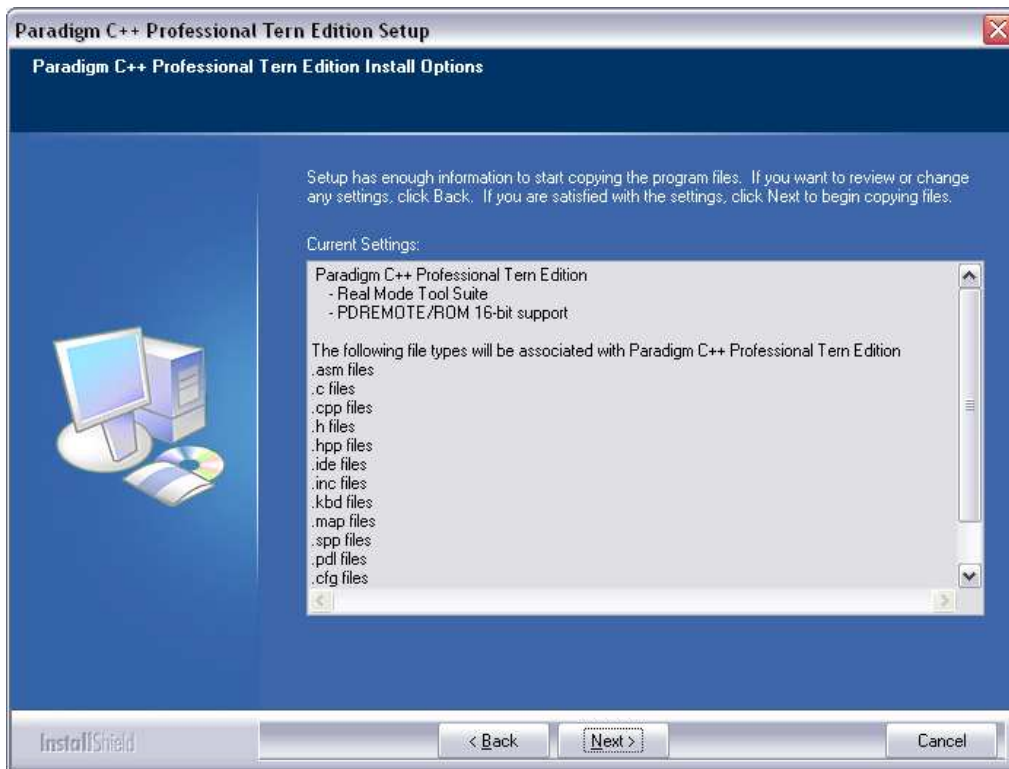
Confirm destination folder and click 'next'



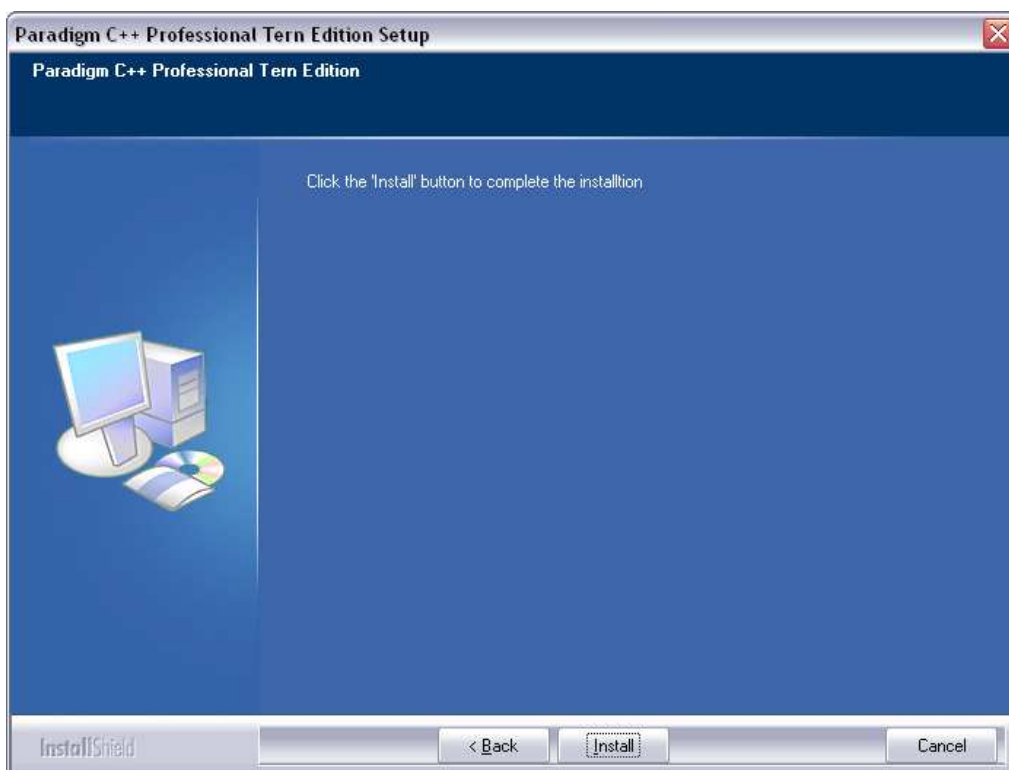
Select the files to associate with Paradigm C++ Professional – TERN Edition and click ‘Next’



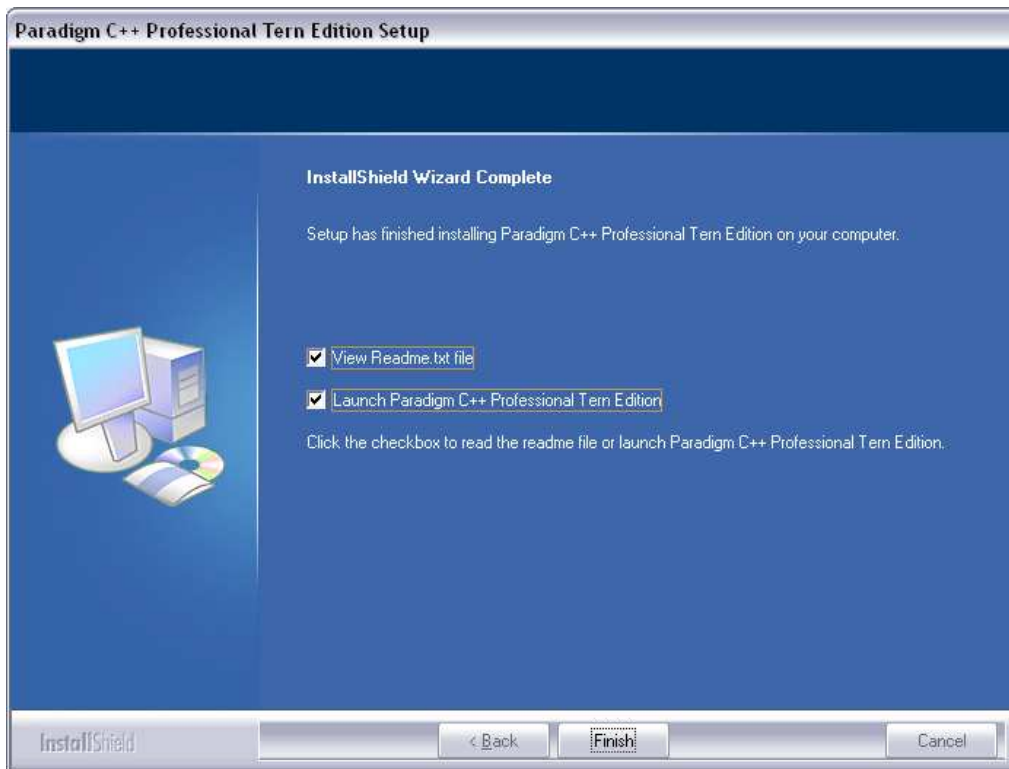
The above options are not required to use Paradigm C++. Click ‘Next’.



Review the install configuration. Click 'Back' to make changes or 'Next' to continue.



Paradigm C++ Professional – TERN Edition is ready to be installed. Click 'Install'.



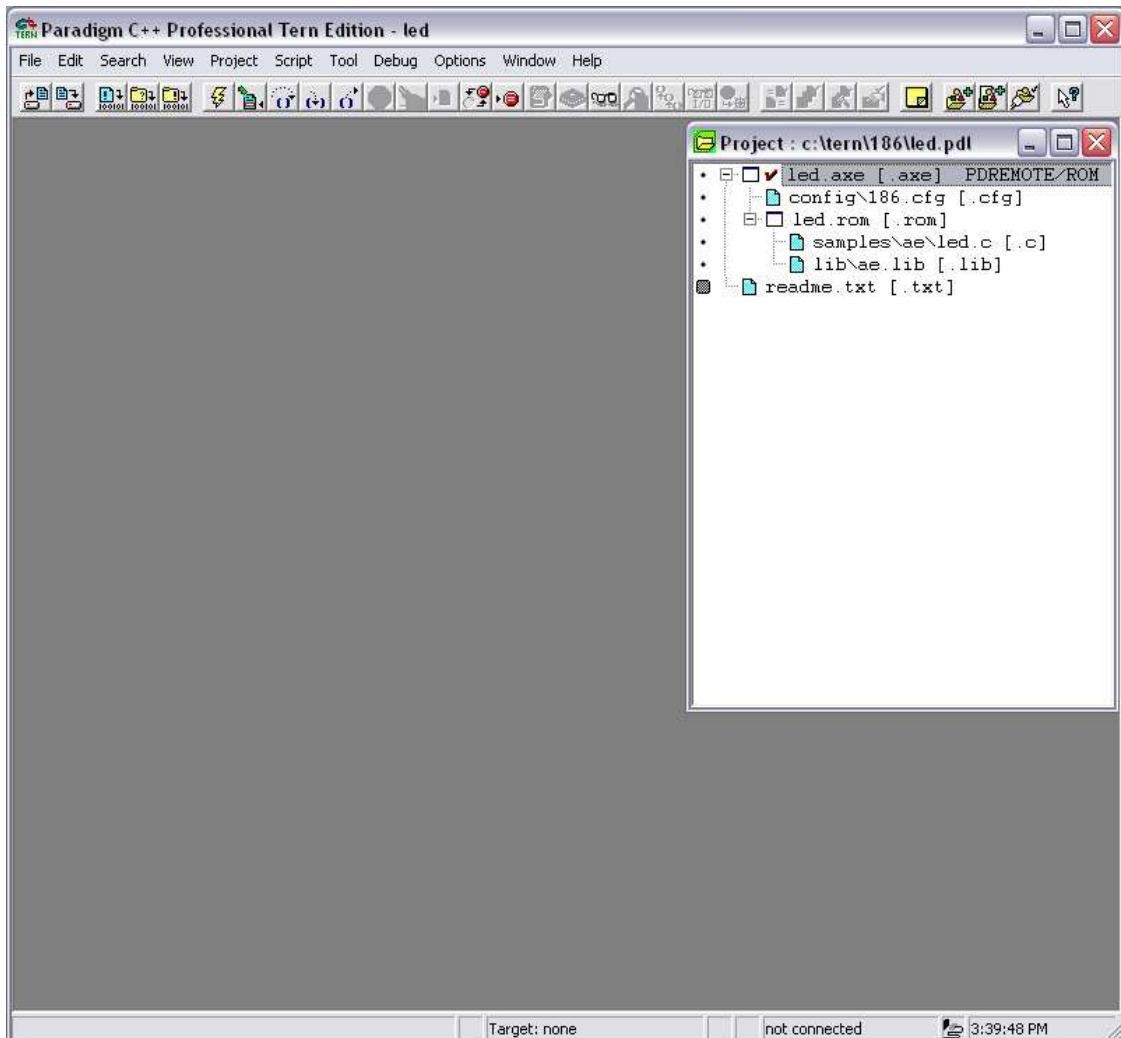
Paradigm C++ Professional – TERN Edition has been installed. Click 'Finish' and the application will run.

## *How do I download and debug an application?*

To download and debug an application you need to:

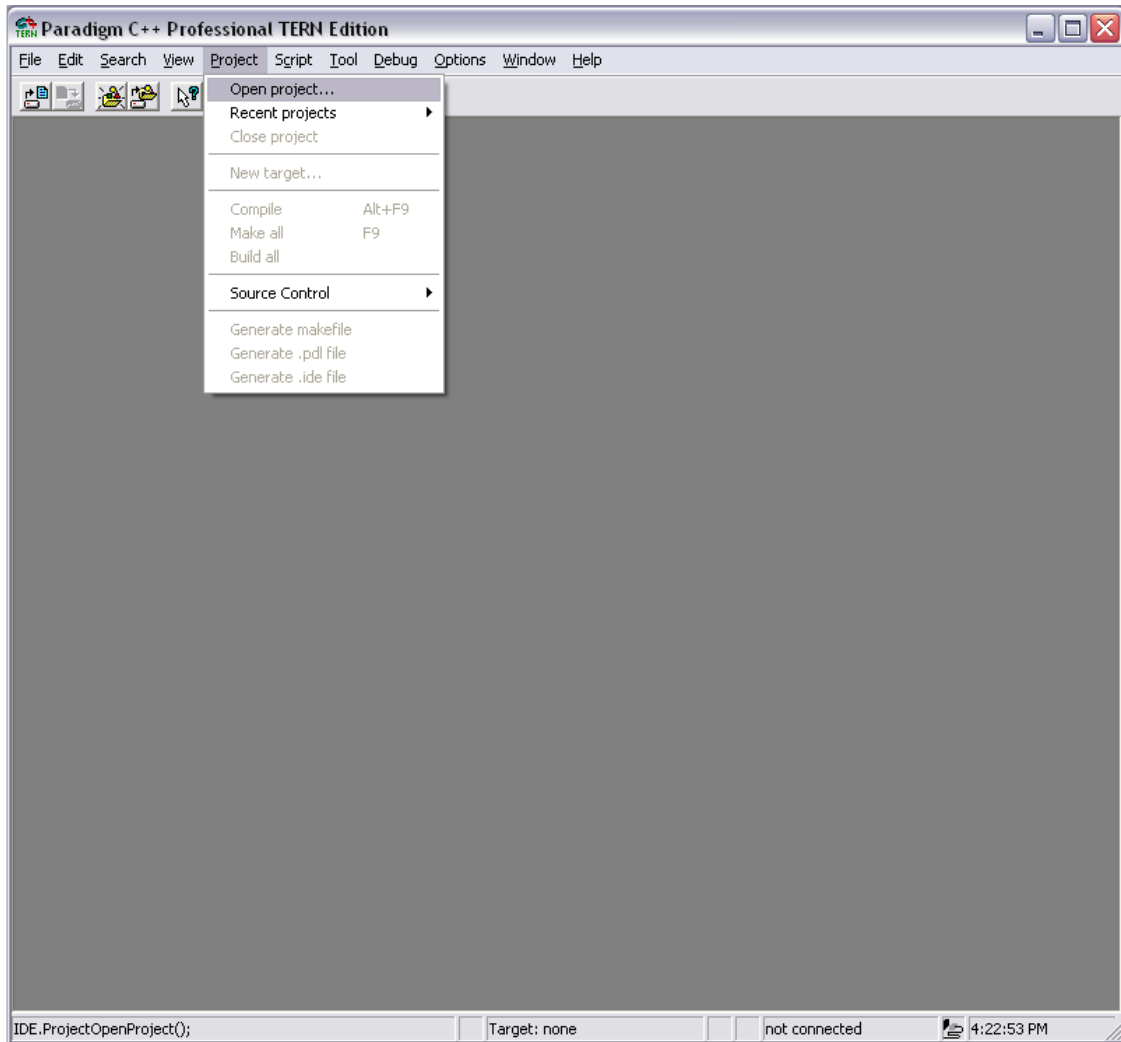
- Open the project you want to run.
- Set the target connection, COM port and baud rate of your PC.
- Build the executable application.
- Download the application to the target device.
- Run the application using the remote debugger.

The next few sections show you how to download and debug the LED sample program for the Am186 platform. The LED sample program will blink the LED on your device. There are LED sample programs for all of the TERN CPU platforms. Most of the steps are identical to the Am186 platform but use a different file directory. See the hardware user guide for specific instructions on running the LED sample program on your device.

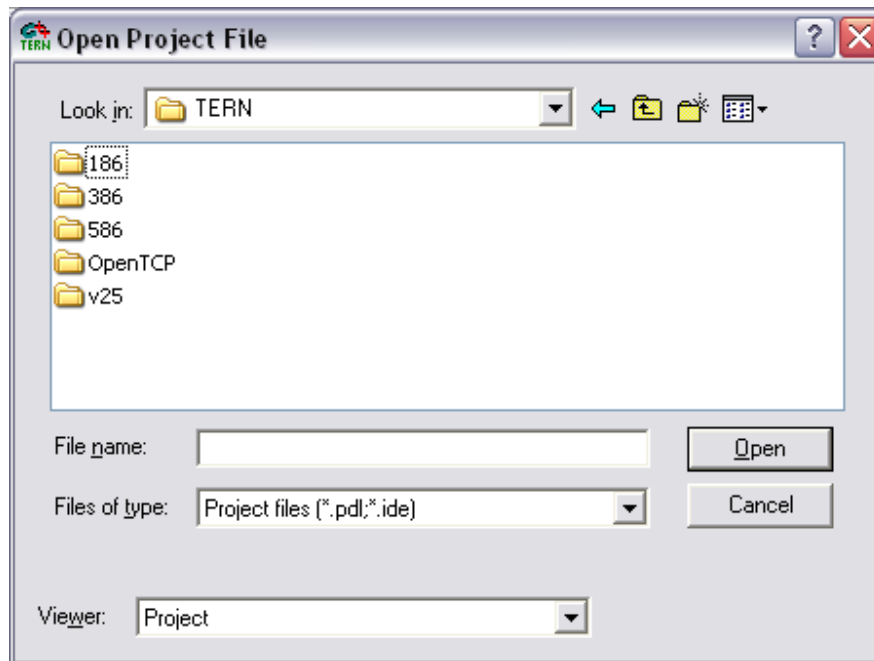


## *How do I open a project?*

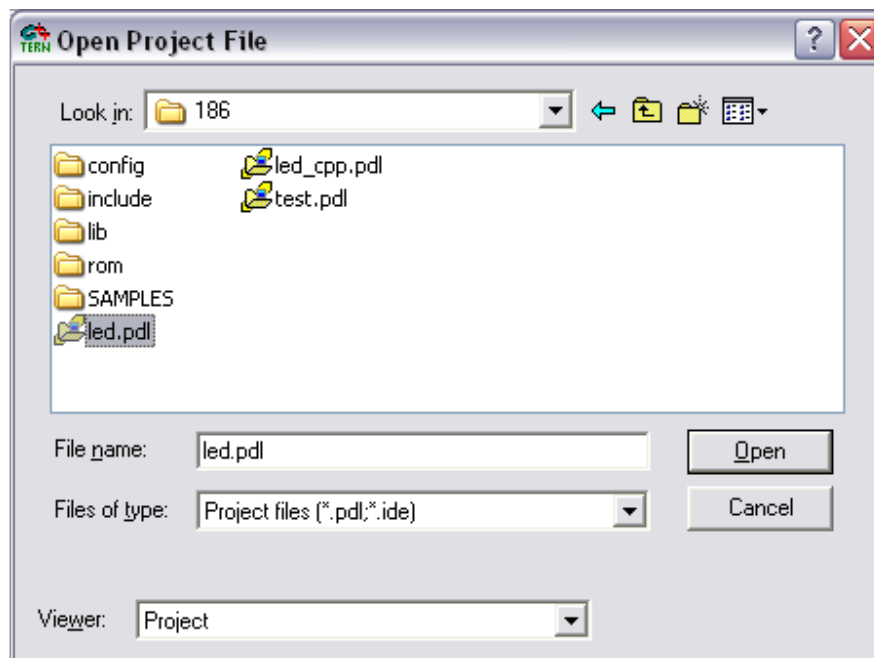
On the tool bar, select “Project” then “Open project”. Projects are organized by CPU type. Am188/Am186 CPU projects are in the C:\TERN\186 directory, i386EX CPU projects are in the C:\TERN\386 CPU directory, SC520/Am586 CPU projects are in the C:\TERN\586 directory, and V25 CPU projects are in the C:\TERN\V25 directory. Projects for Paradigm C++ Professional – TERN Edition have the suffix “.pdl”.



Find the directory that matches your hardware.

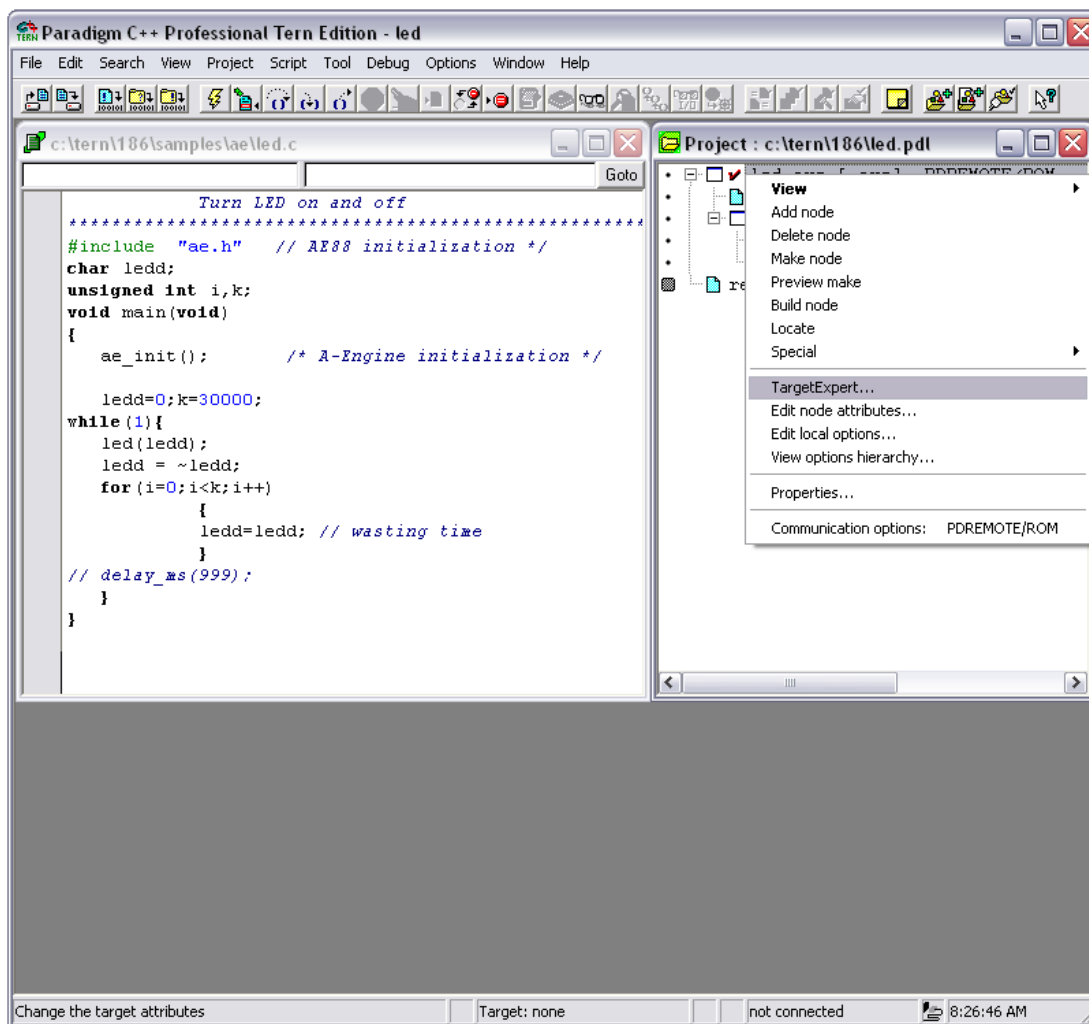


Select the project you want to open. For this example, we'll open the “led.pdl” sample project.



## *How do I configure the target connection?*

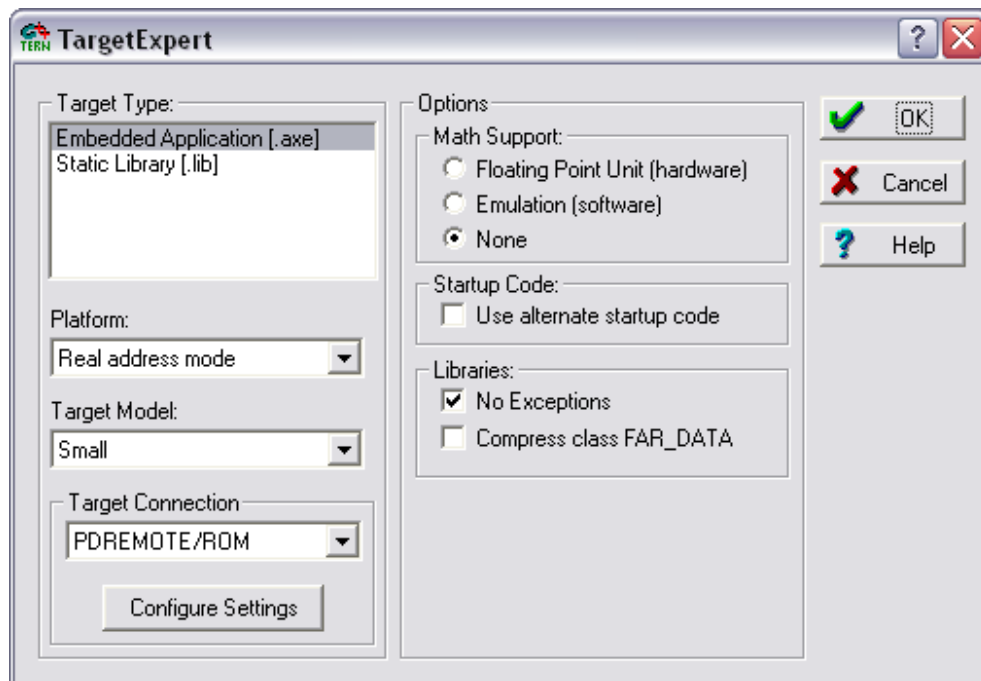
Use the TargetExpert to change the target connection type, baud rate and COM port used to communicate with the target device. Right click on the [.axe] node you want to set the baud rate for and select 'TargetExpert' on the popup menu.



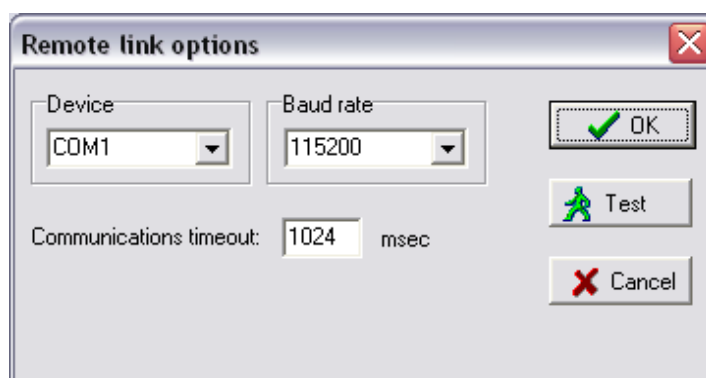


On the TargetExpert window, set the “Target Connection” to “PDREMOTE/ROM”. This will instruct Paradigm to generate an “.axe” file that can be downloaded into non-volatile SRAM of your device and run using the remote debugger. Double check that the “Target Type” is set for “Embedded Application” and “No Exceptions” box is checked under “Libraries”.

Next, select “Configure Settings” to set the COM port and baud rate of your PC.



Select the COM port your PC uses and the baud rate your target device communicates at. Most TERN devices communicate at 115200 baud, but some run at slower rate. See your device’s user manual for details.

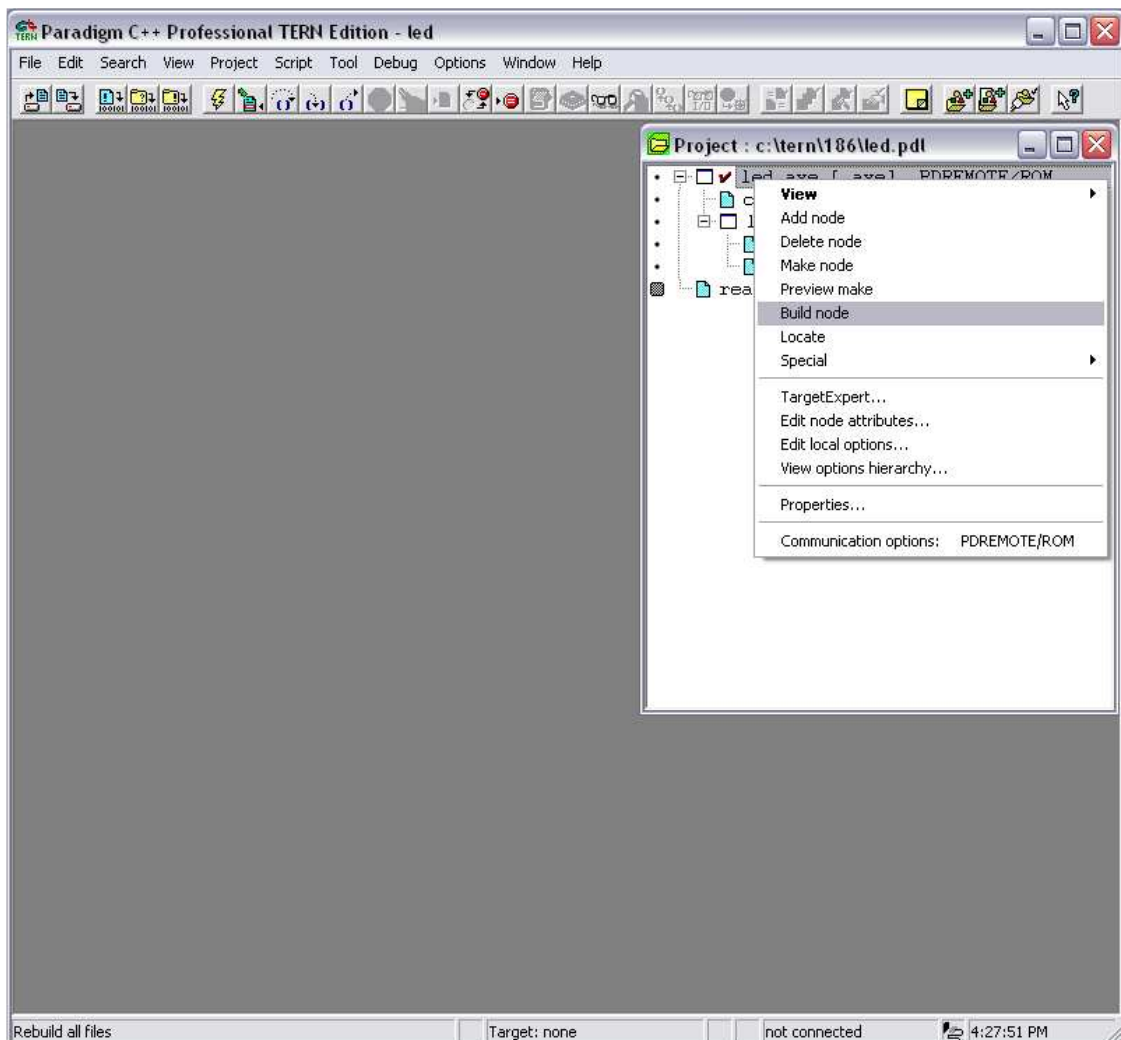


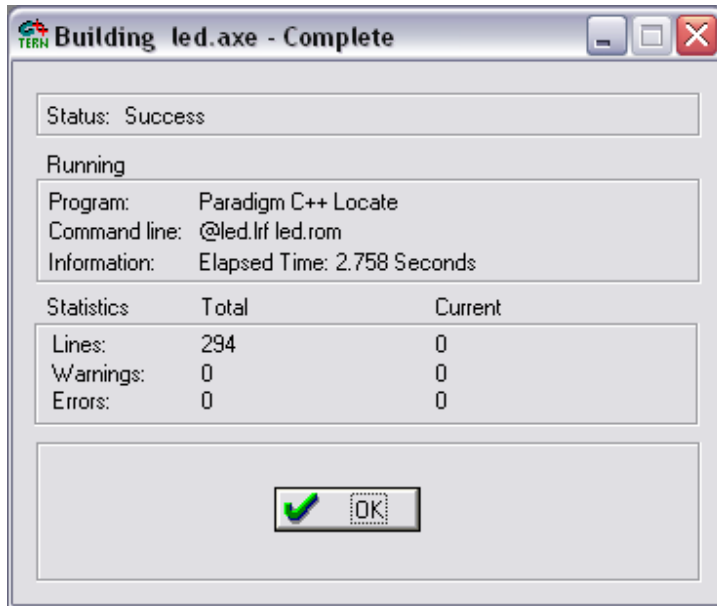
## *How do I build a project?*

To build a project, right click the [.axe] node you wish to build and select “Build node”. Paradigm will compile and link all the child nodes under the [.rom] node and produce a [rom] file. It will then locate the .ROM code to the correct memory space based on the configure file in the [.cfg] node.

When the target is set for 'PDREMOTE/ROM', the final output file will be an .axe file. The .axe file is used for remote debugging with your target device.

When the target is set for ‘No Target/ROM’, the final output file will be a HEX or BIN file (depending on the .cfg file settings). The HEX or BIN file is used to burn your application into Flash or EPROM. The full development version of Paradigm is required to perform this operation.

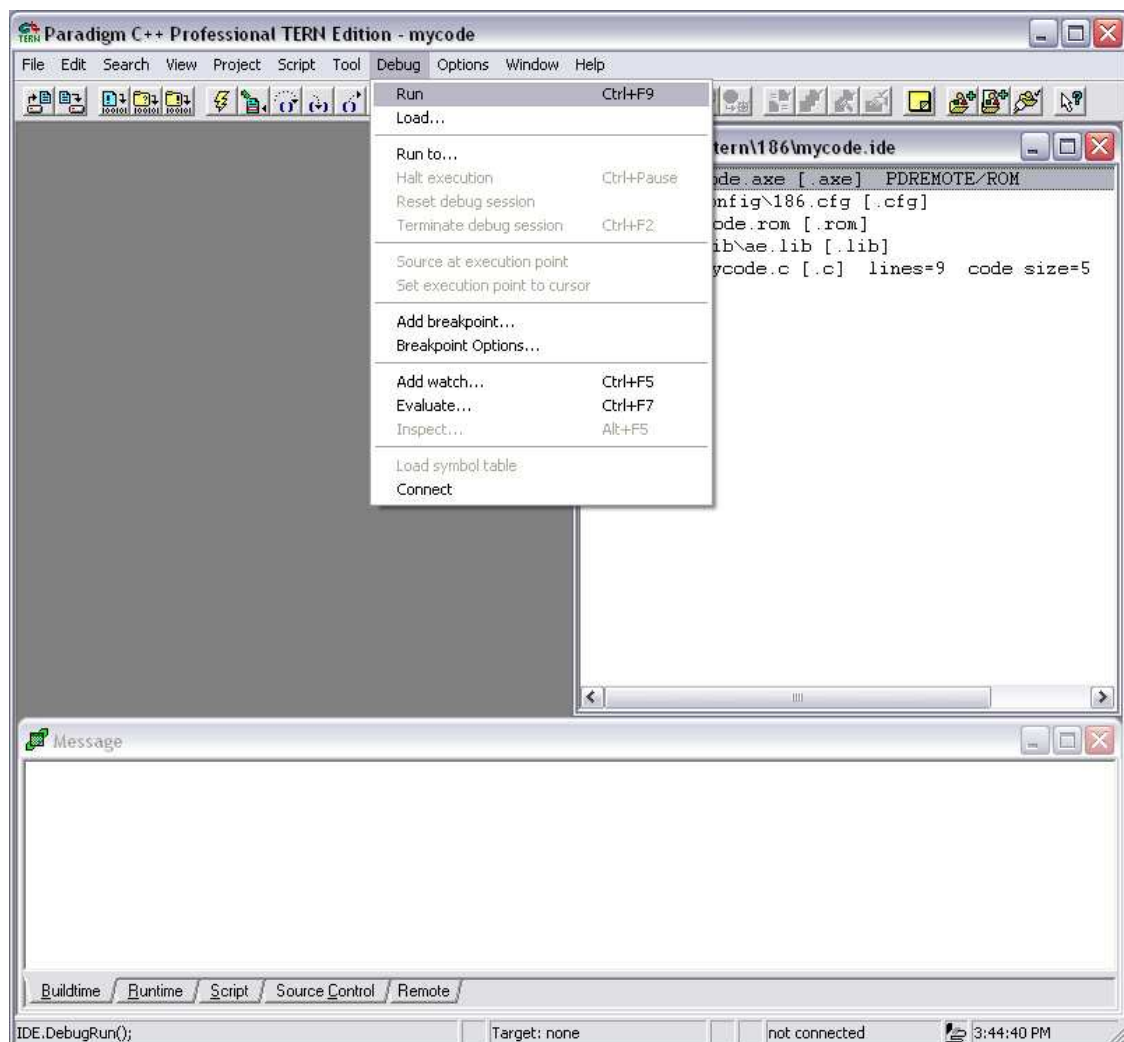




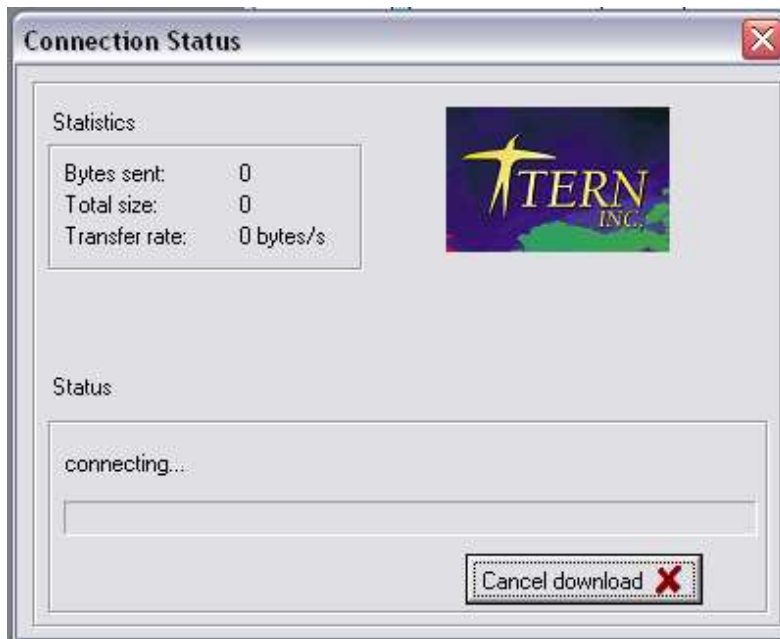
## *How do I download a project to my device?*

Once you have built the code it is ready to download to your device. There are several ways to download an application to a device.

- Press Ctrl+F9 or select 'Run' from the debug menu to download the application and immediately run it.
- Double click the [.axe] node to build and download the project. The debugger will halt execution at the beginning of the code.
- Click the STEP OVER button to build and download the project the same as double clicking the [.axe] node.



The connection status window will open and show the connection and download status.



## *What do I do if I can't connect to the target device?*

If you can't connect to the target device, check that:

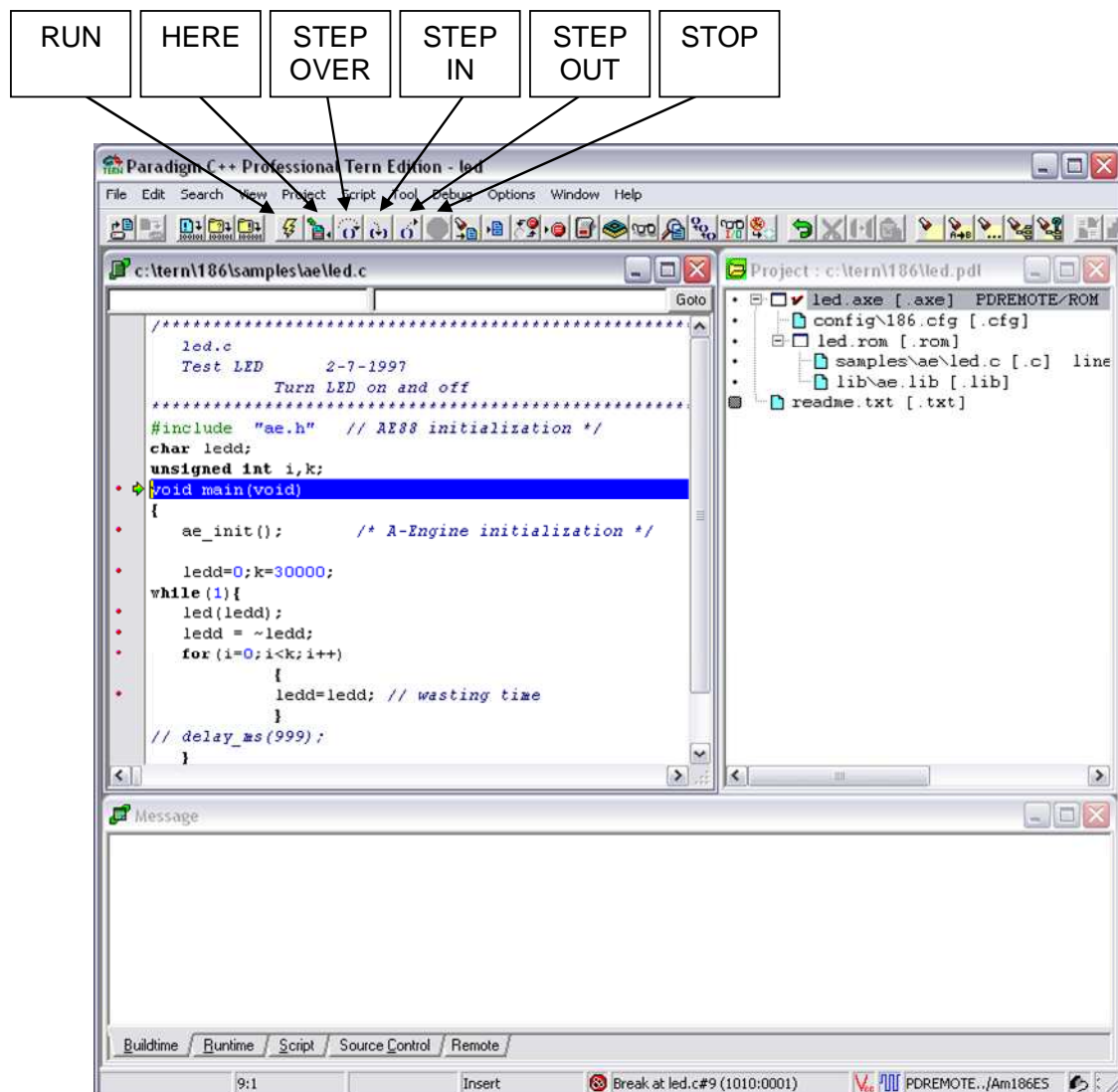
- The RS-232 cable is connected to the SER0 header of your device.
- The red stripe of the serial cable points to pin 1 of the serial header.
- The LED blinks twice then stays lit when you power on the device.
- The baud rate is correct for your device.
- The COM port is correct for your PC.

**CHECK THE HARDWARE USER GUIDE FOR DETAILS ON CONNECTING TO YOUR PARTICULAR DEVICE.**

## *How do step through the code?*

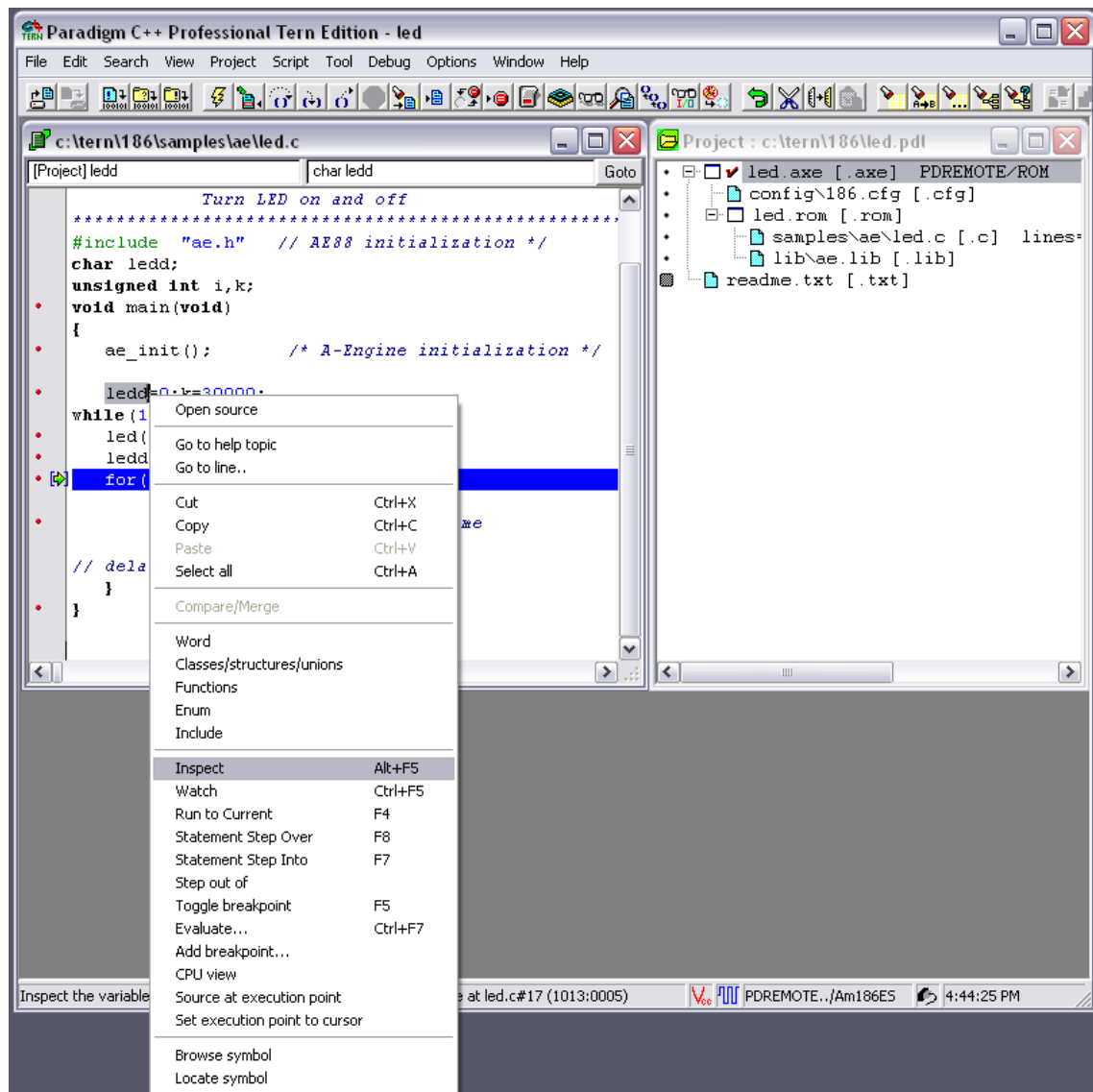
The remote debugger allows you to run and step through your code in different ways.

- **RUN:** Click the “Run” button to run the code uninterrupted.
- **STOP:** Click the “Halt execution” button to stop code that is running.
- **HERE:** Click the ‘Run to here’ button to run an application to a specific line in the code. Place the cursor to the desired run-to line before clicking ‘Run to here’.
- **STEP OVER:** Click ‘Statement step over’ to execute a statement like a function call without entering the code of the function.
- **STEP IN:** Click ‘Statement step into’ to enter the source code of a function.
- **STEP OUT:** Click ‘Step out of’ to leave the current function and return to the calling code.

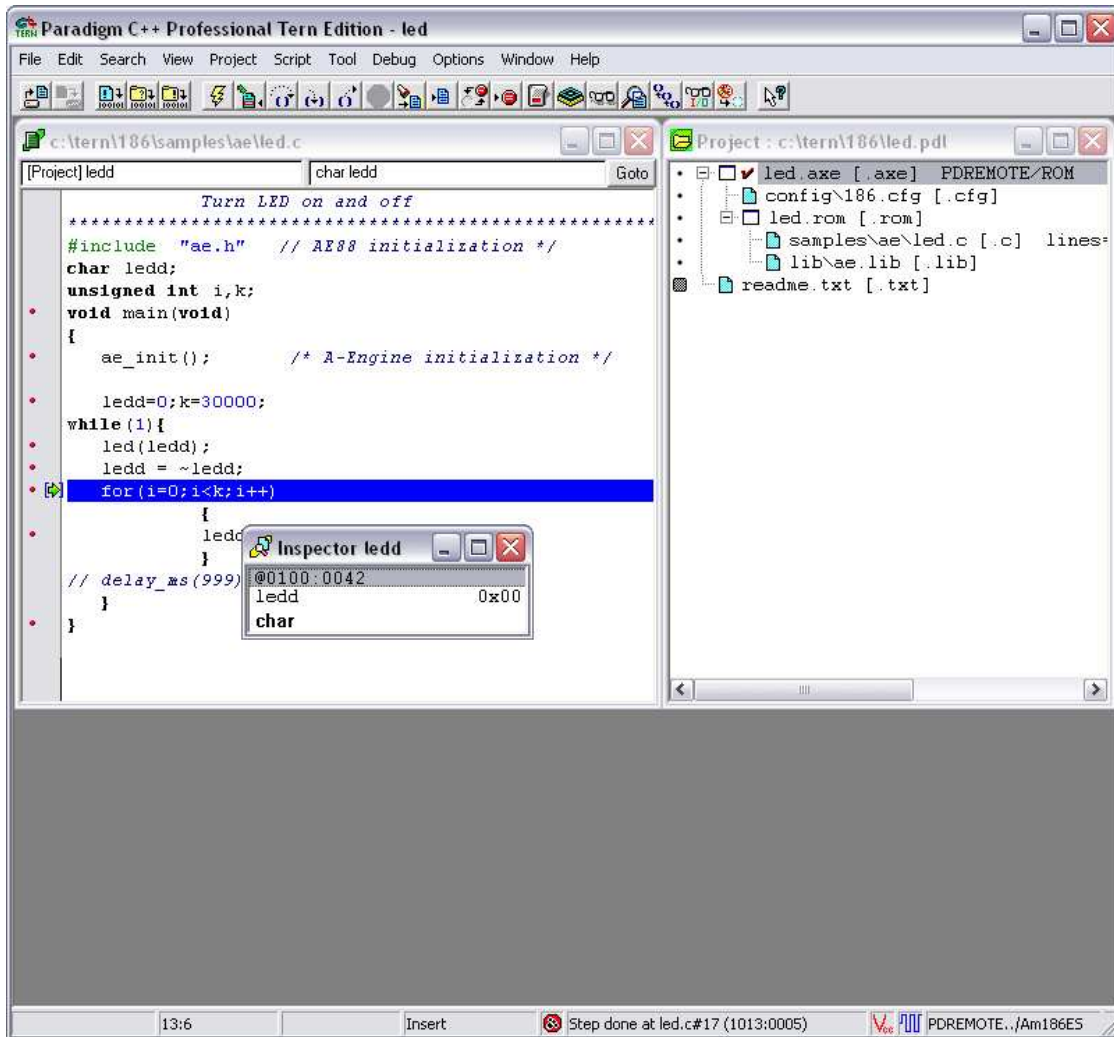


## *How do I inspect the value of a variable?*

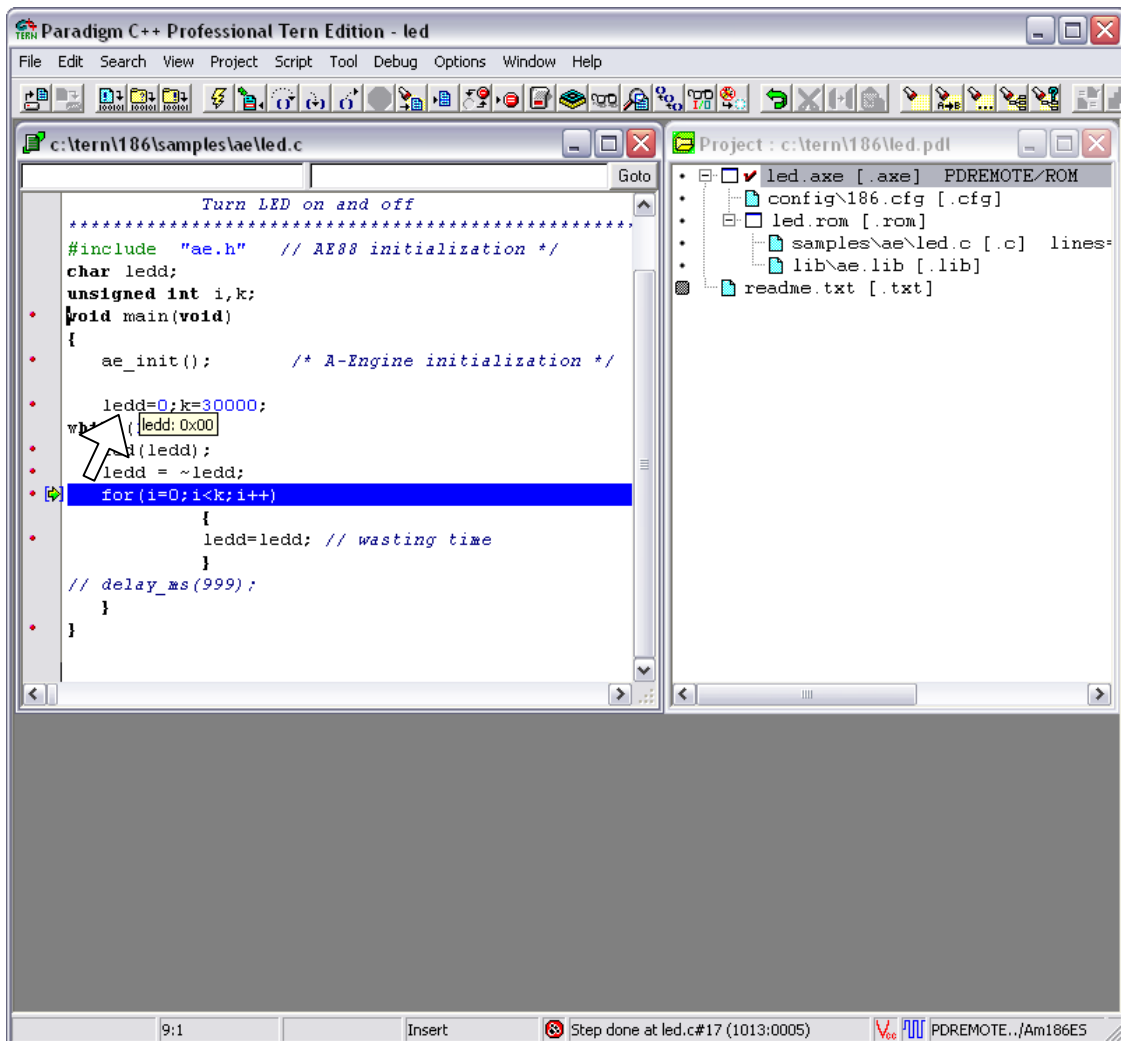
After downloading the code or during debugging, if you want to see what value of the variable `ledd`, you can simply use the mouse to select the variable and then right click on it. Select “Inspect” from the menu that appears after right clicking `ledd`.



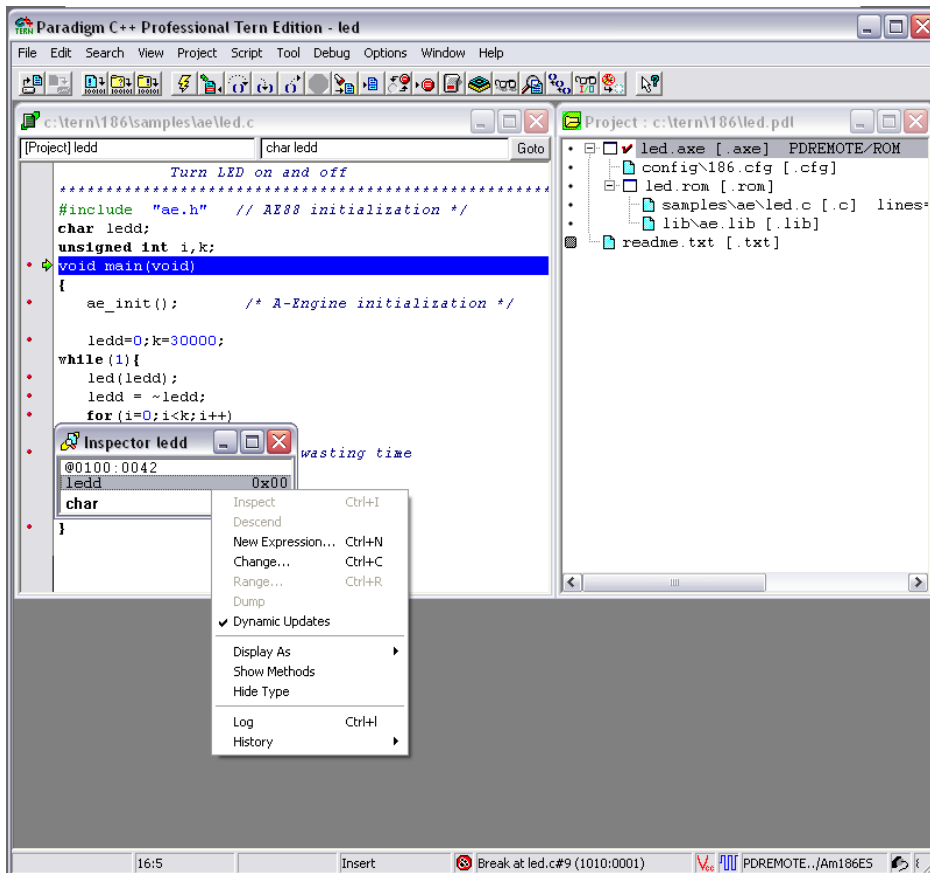




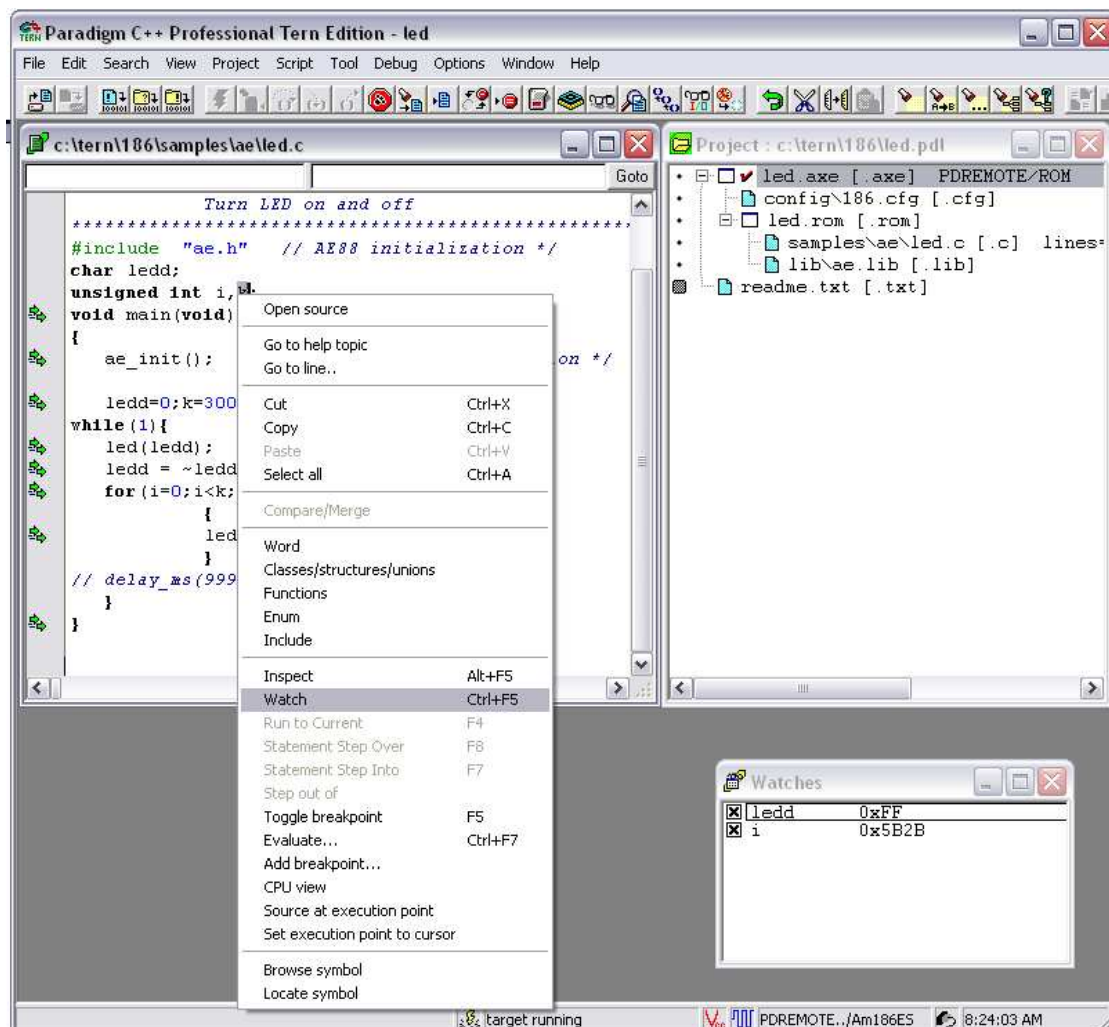
Also, you can move the cursor over a variable name and the current value will pop up.



To watch a variable's value while your code is running, be sure the dynamic updates are enabled. You can enable dynamic updates by right clicking the 'Inspector' window and select 'Dynamic Update' from the pop up menu.

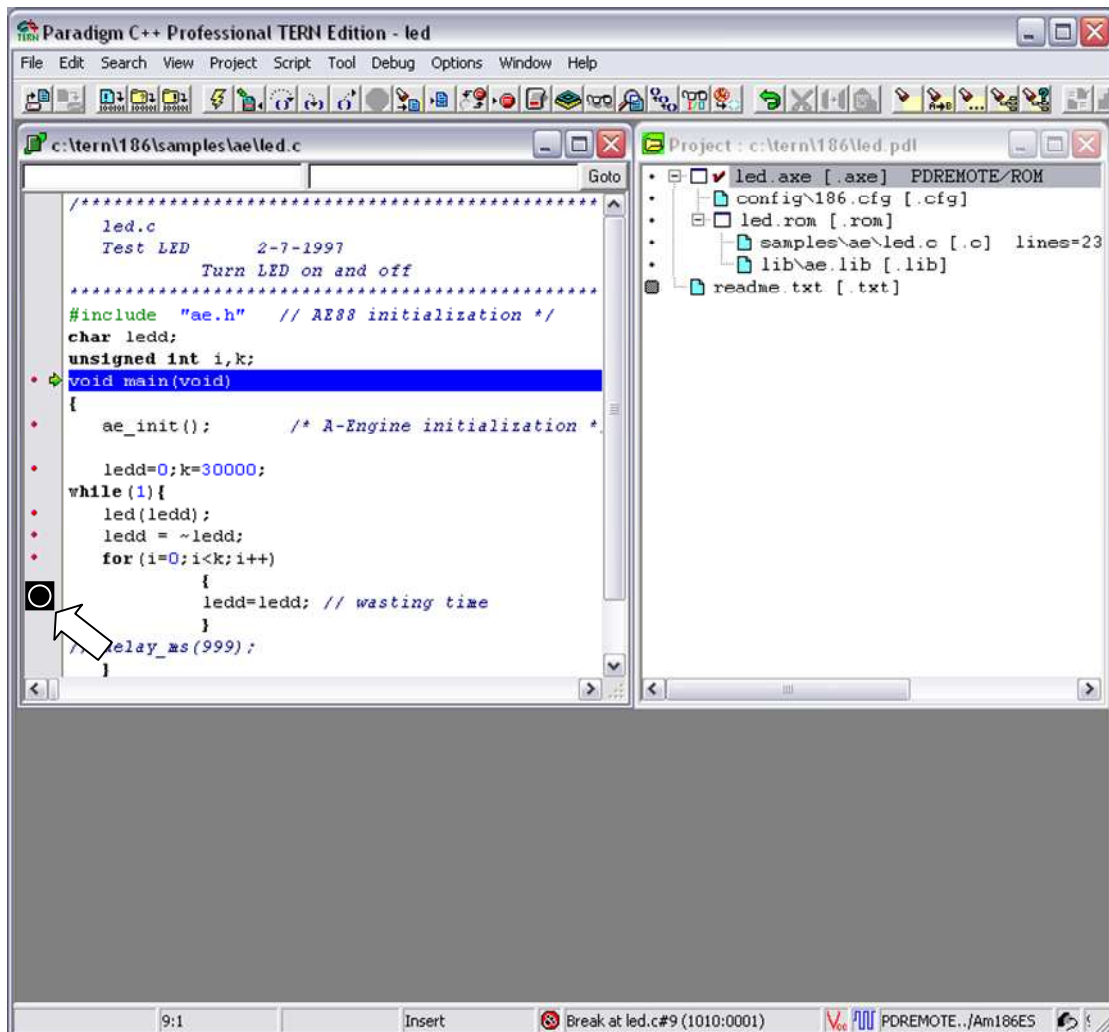


Inspecting a variable allows you to view it in its own window. To view multiple variables in a single window, use the 'Watch' function. Right click on the variable you want to add to the watch list and select 'Watch' from the popup menu.

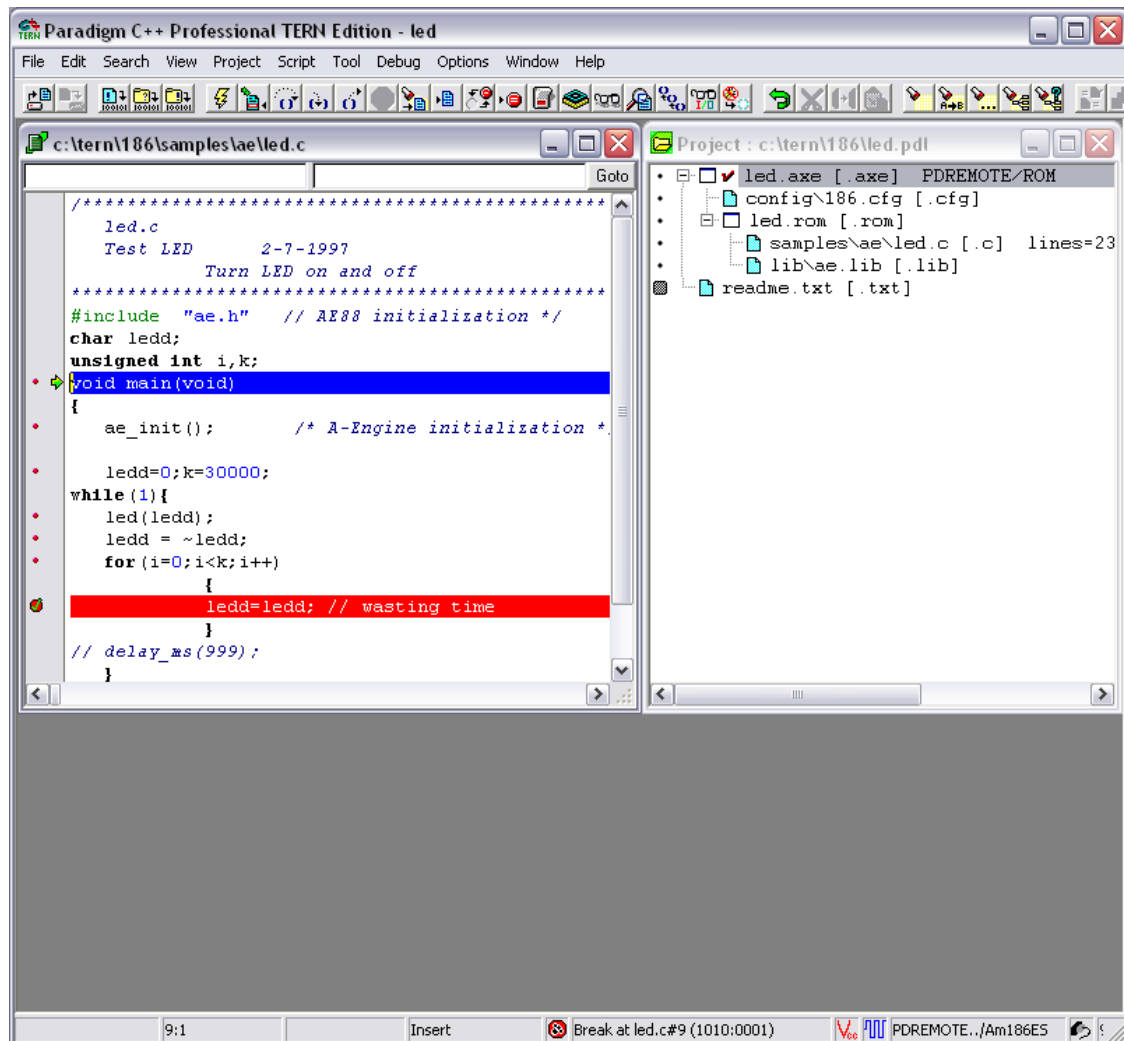


## *How do I set a breakpoint?*

Breakpoints cause the debugger to halt execution at specific lines of your code. To set a breakpoint, move the mouse to the left margin on your source code and click the on the line you want the code to halt at. To remove the breakpoint, click again on the margin where the breakpoint is set.

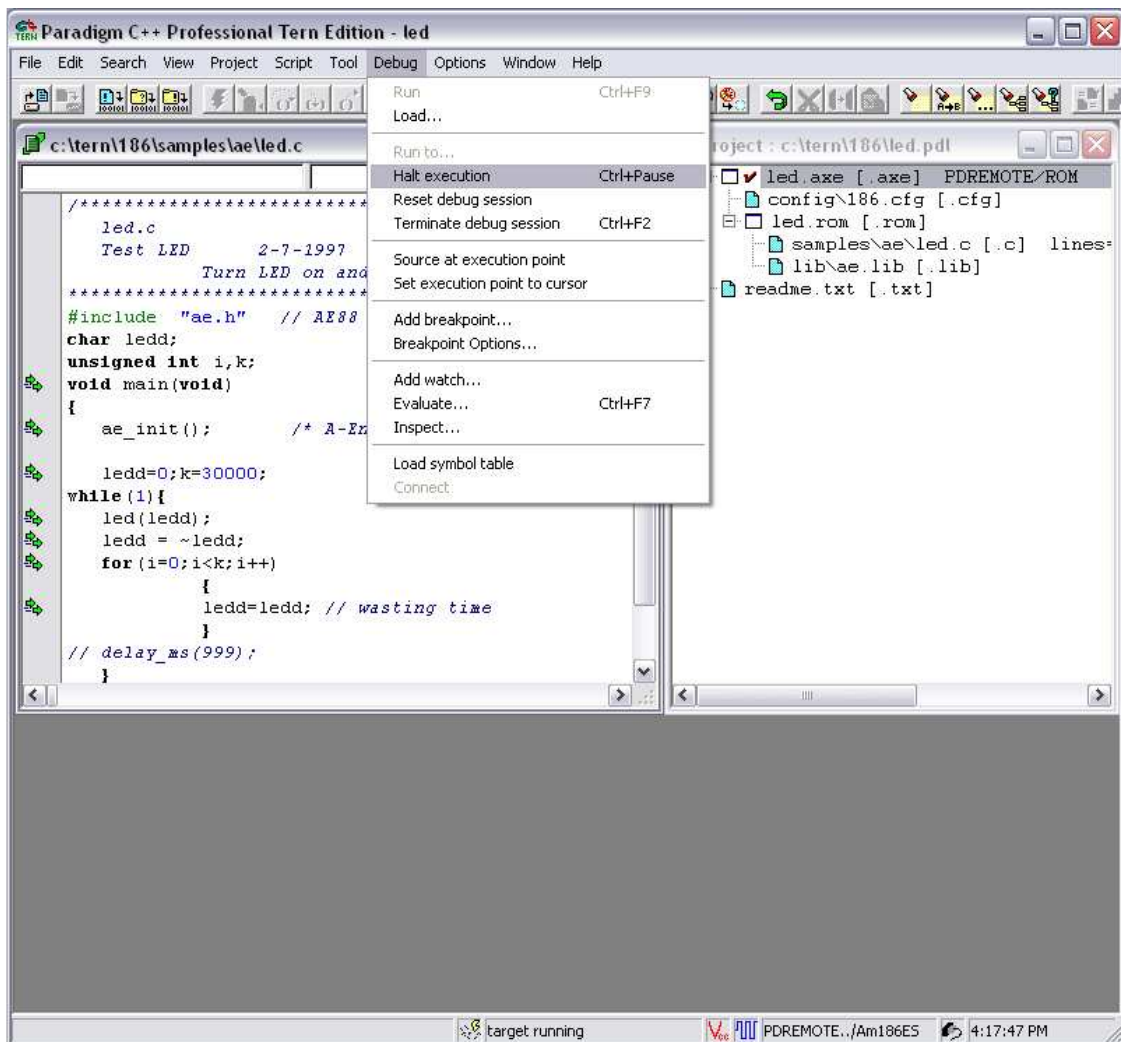


The breakpoint is marked with a red horizontal bar.



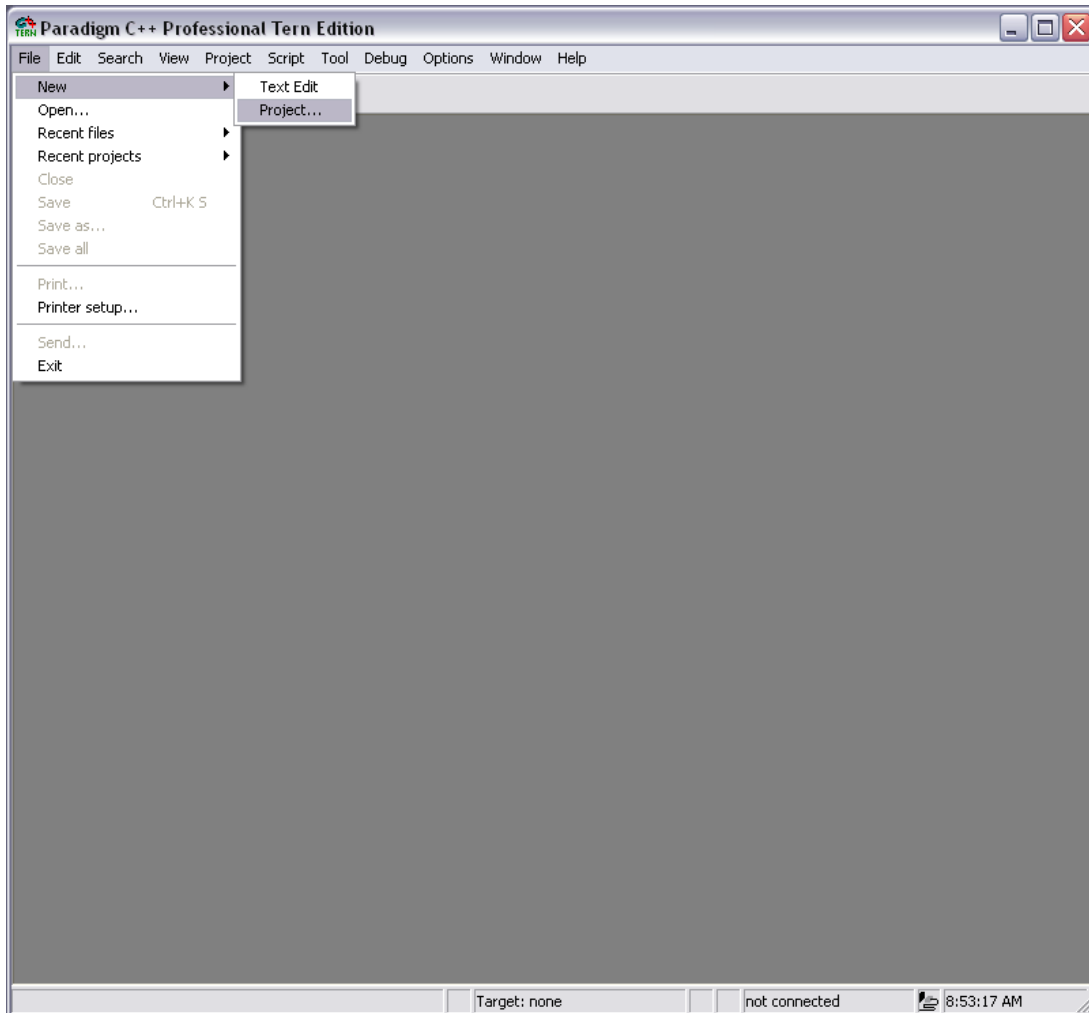
## *How do I terminate the debug session?*

Select the 'Debug' from the top menu bar, click "Halt execution" then click "Terminate debug session", or press "Ctrl+Break" then press "Ctrl+F2" keys. The led on your controller should blink twice then stay on indicating it has been RESET.



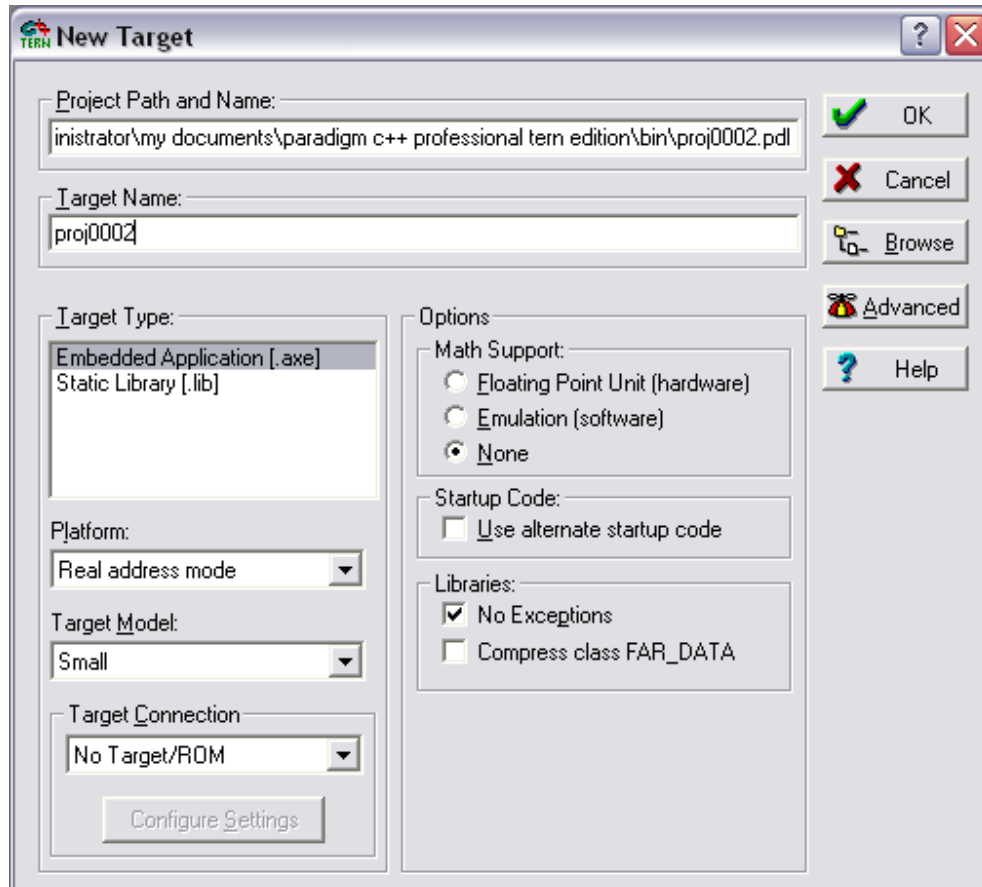
## *How do I make a new project?*

To create a new project, select 'File', 'New' then 'Project' from the main menu.



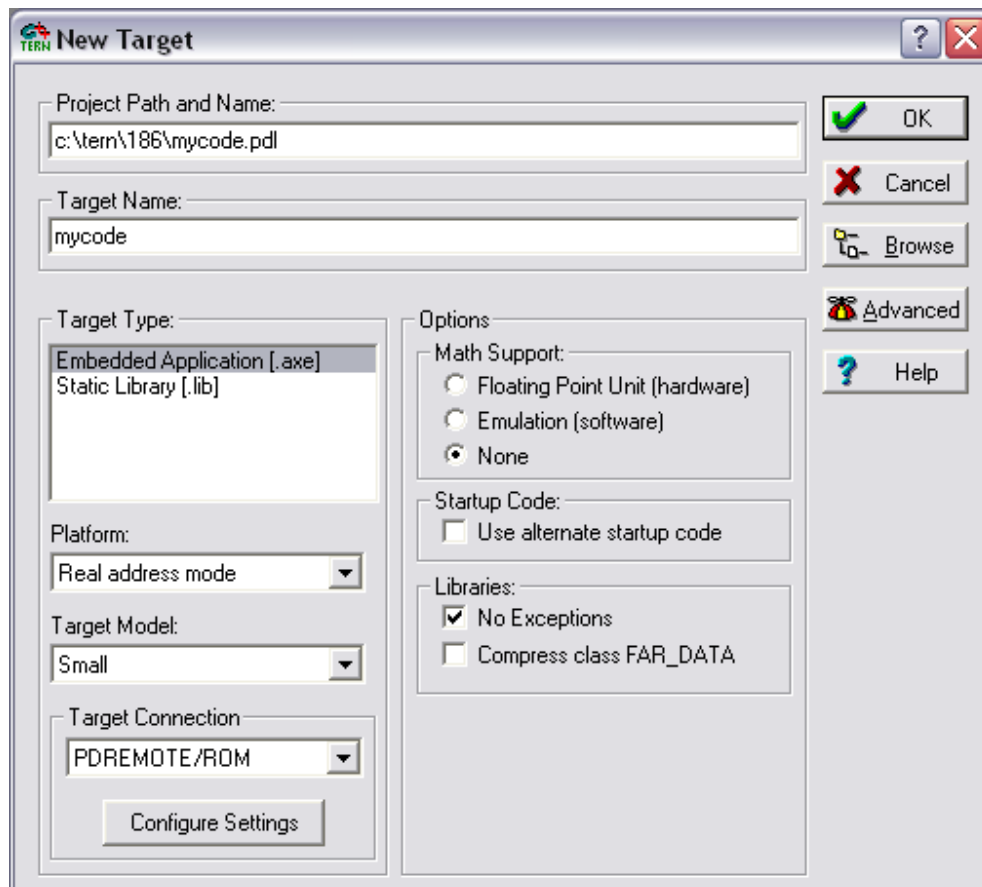


The 'New Target' window will open and allow you to name the project and select the desired folder to put it in.

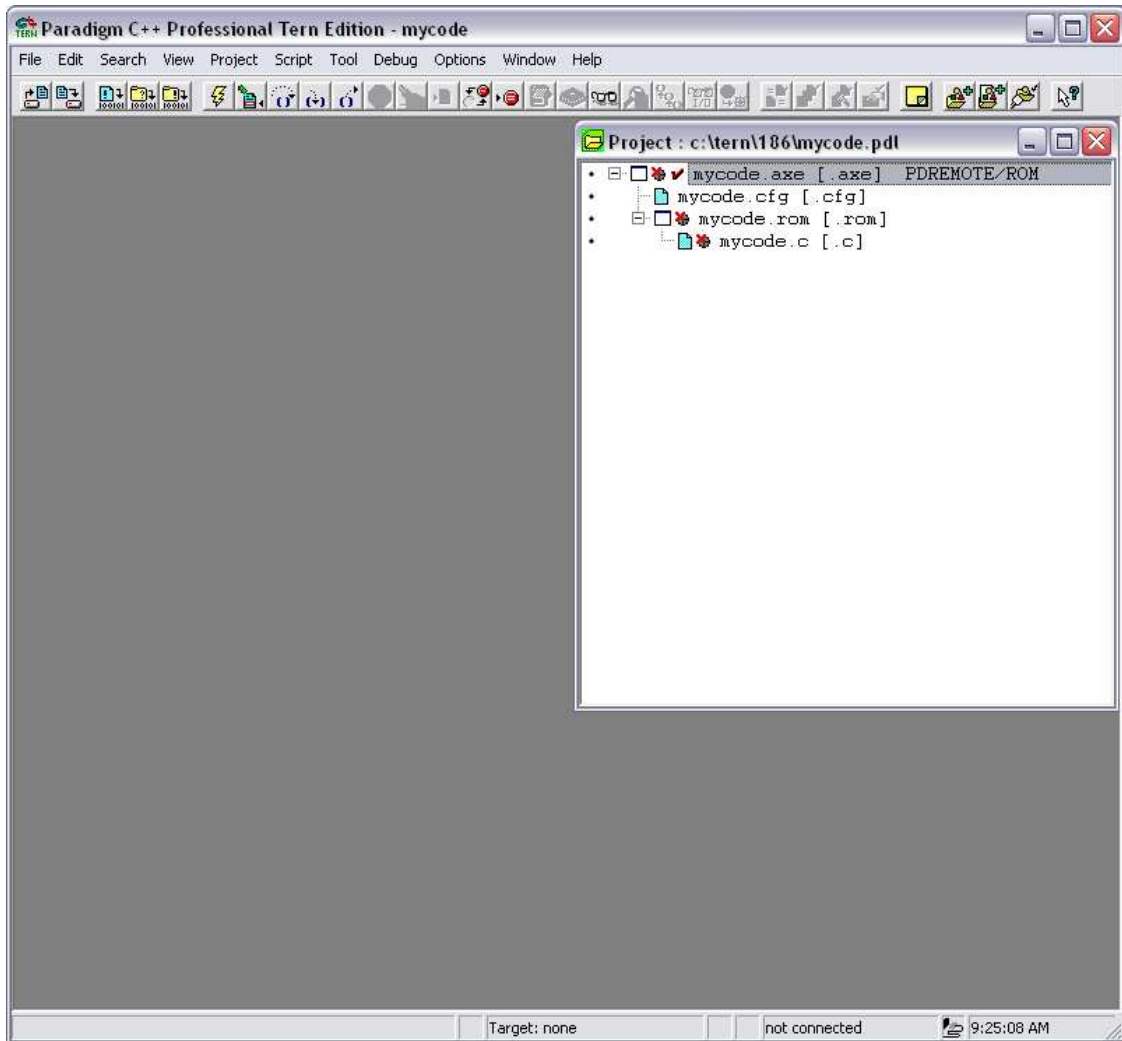


Set the path to c:\tern\186 for 186 and 188 based devices, c:\tern\386 for 386 devices, c:\tern\586 for 586 devices or c:\tern\V25 for V25 devices. Make sure the 'Libraries: No Exceptions' is **CHECKED**.

You also need to define the Target Connection as 'PDREMOTE/ROM' for debugging. Selecting 'No Target/ROM' will generate a BIN or HEX file that will be used to burn your code into EPROM or FLASH. You must have the full version Paradigm C++ Professional-TERN Edition, not the Evaluation version, to generate BIN/HEX files.



After clicking 'Ok' on the 'New Target' window, a new project is created.



## *What is a node?*

In the Project window, each line in the project tree is a ‘node’. The root node is always the [.axe] node, which represents the final application code ready for downloading to your controller. The [.axe] node has two children nodes: [.cfg] and [.rom].

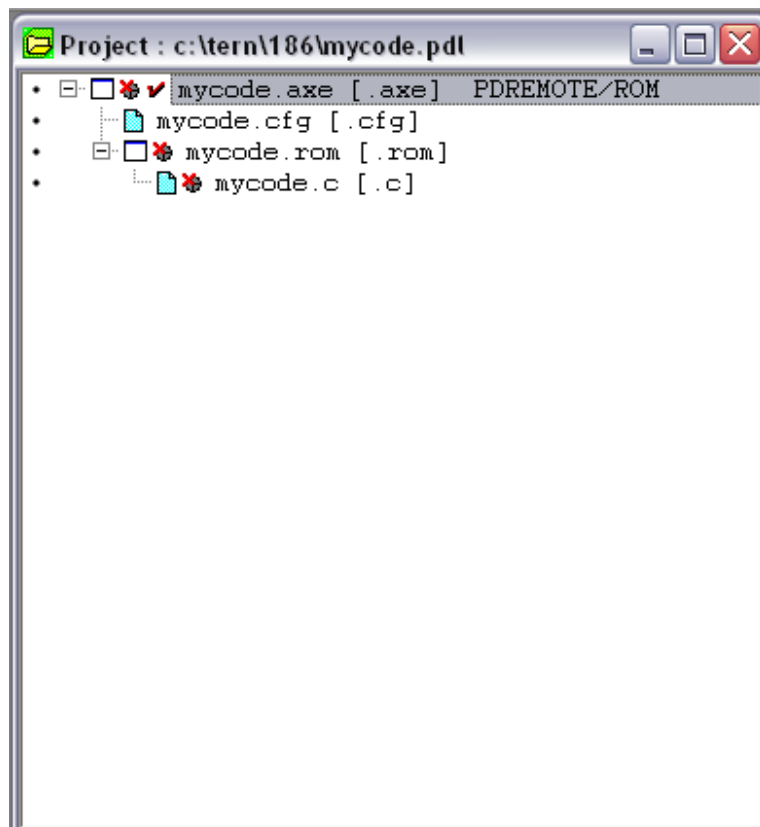
The generation of the [.axe] nodes depends on the [.cfg] node and the [.rom] node.

The [.cfg] node is the configuration file for mapping your code and data. The [.rom] node is the product of the application source compiled and linked by C compiler and linker.

The default configuration file is under c:\tern\186\config\186.cfg. You may use it by change the name of the “mycode.cfg”.

The mycode.rom [.rom] node has its own children. The numbers of children nodes depends on your application including all source files and necessary libraries.

The node “mycode.c” may not be sufficient to create an application. You may need to add supporting files and libraries like “c:\tern\186\lib\ae.lib” to mycode.rom. See below for details on adding nodes.

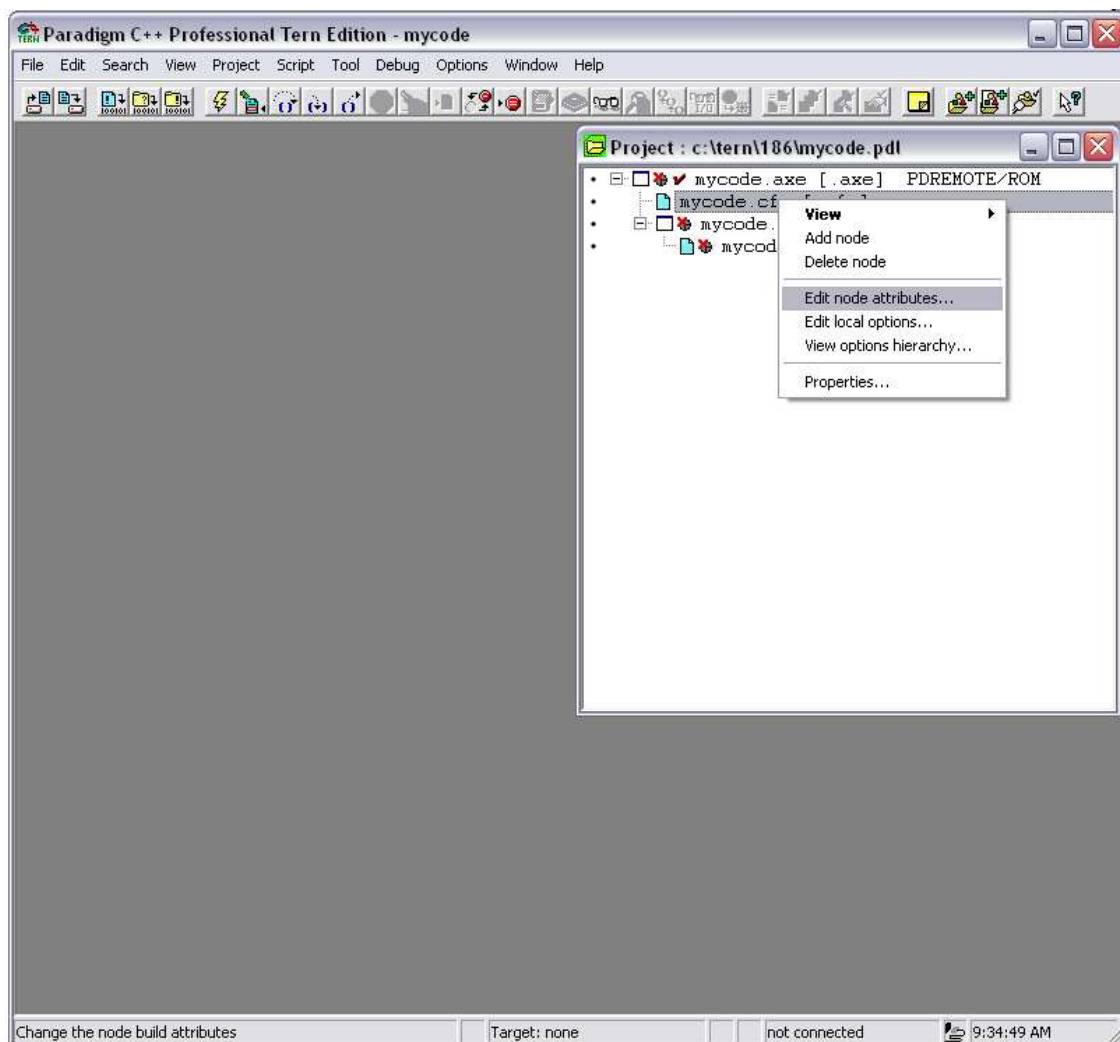


## *How to move a node in the Project window?*

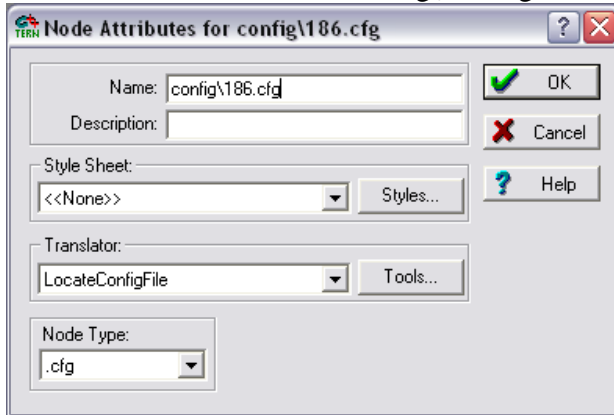
Highlight the node you want to move, hold down the “Alt” key while pressing the up, down, left or right arrow depending on where you want to place the node. You may also drag the node with the mouse and drop it on the new parent node.

## *How to change, rename, or delete a node?*

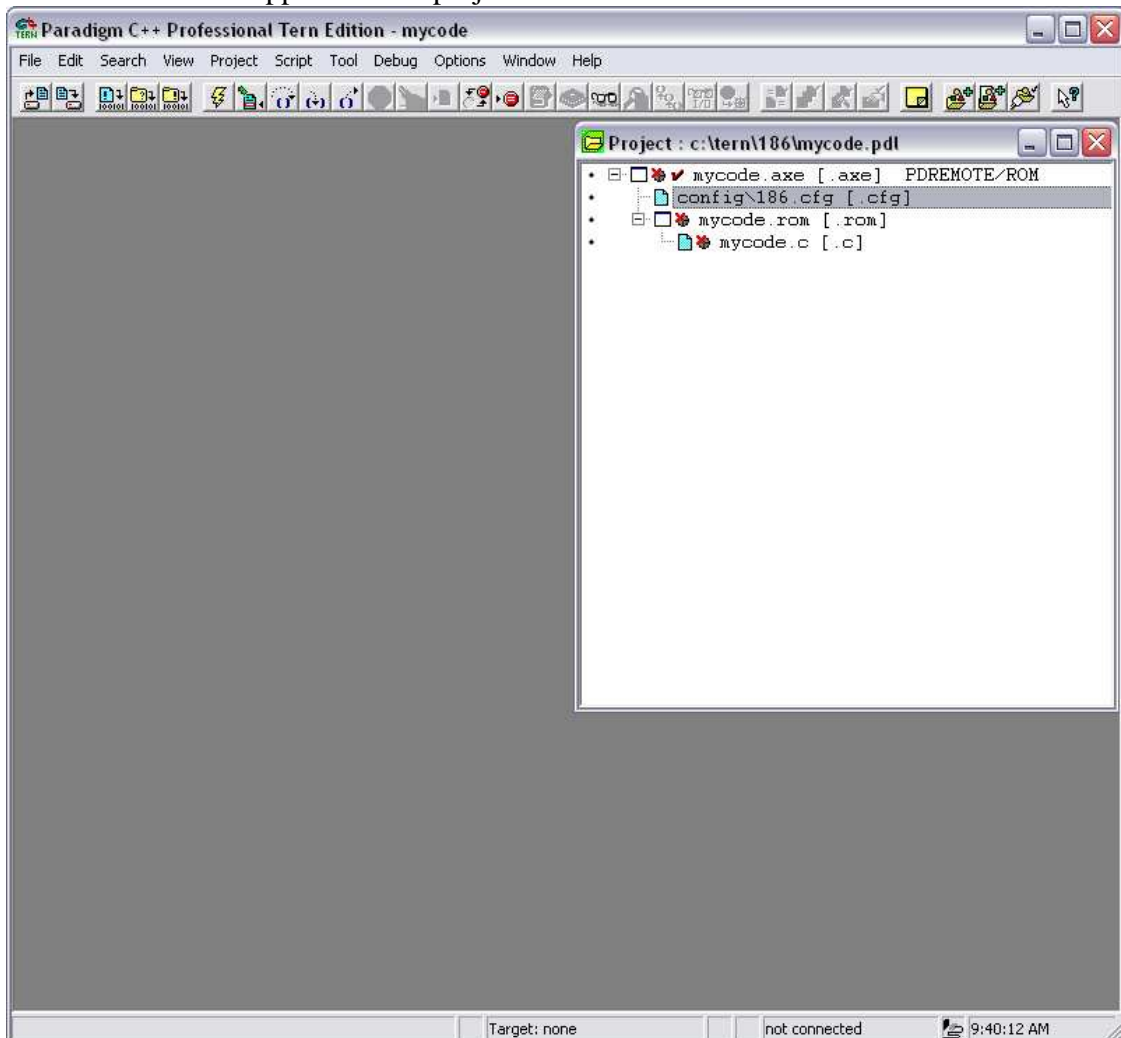
To rename the “mycode.cfg” node to “config\186.cfg”, right click the “mycode.cfg” node and select “Edit node attributes”.



Rename the node name to “config\186.cfg” and click ‘Ok’.

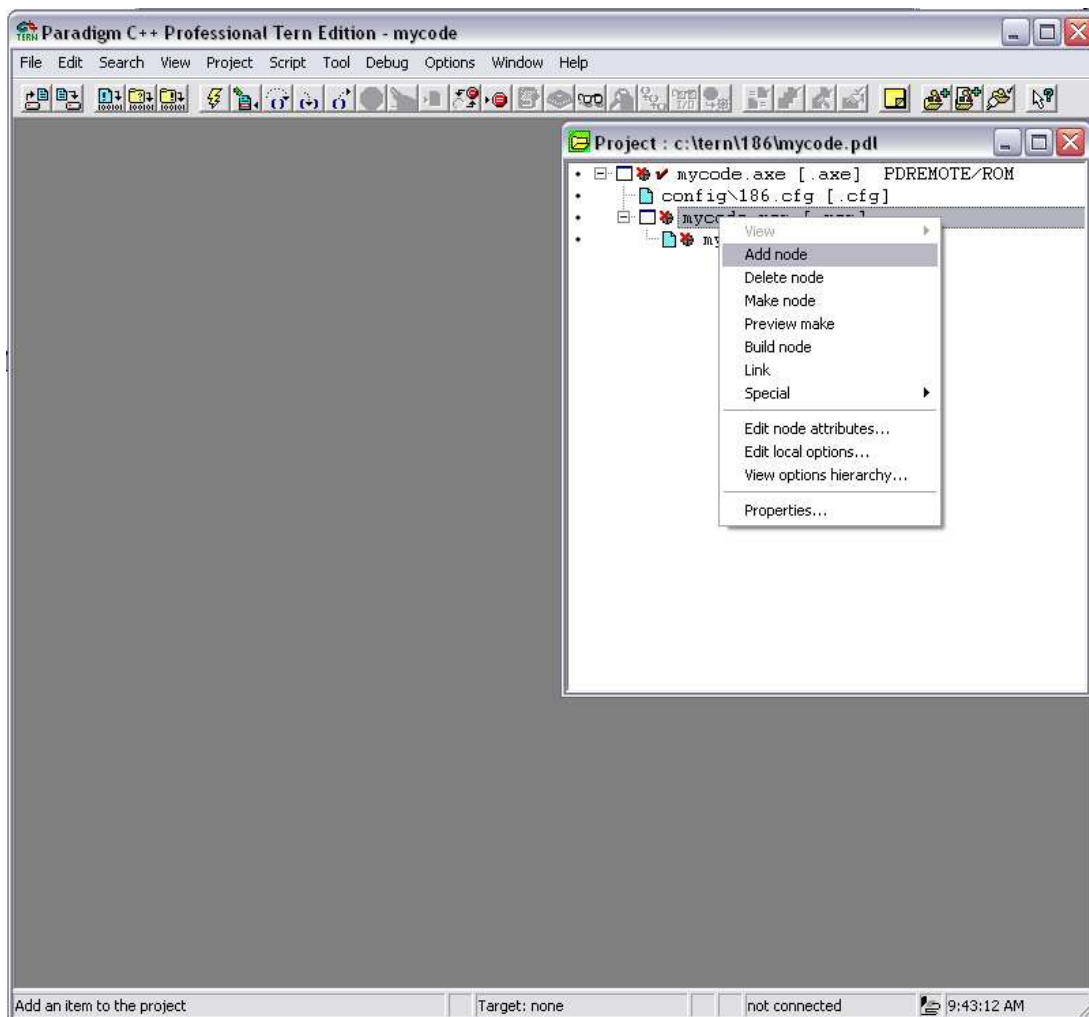


The new node now appears in the project tree.

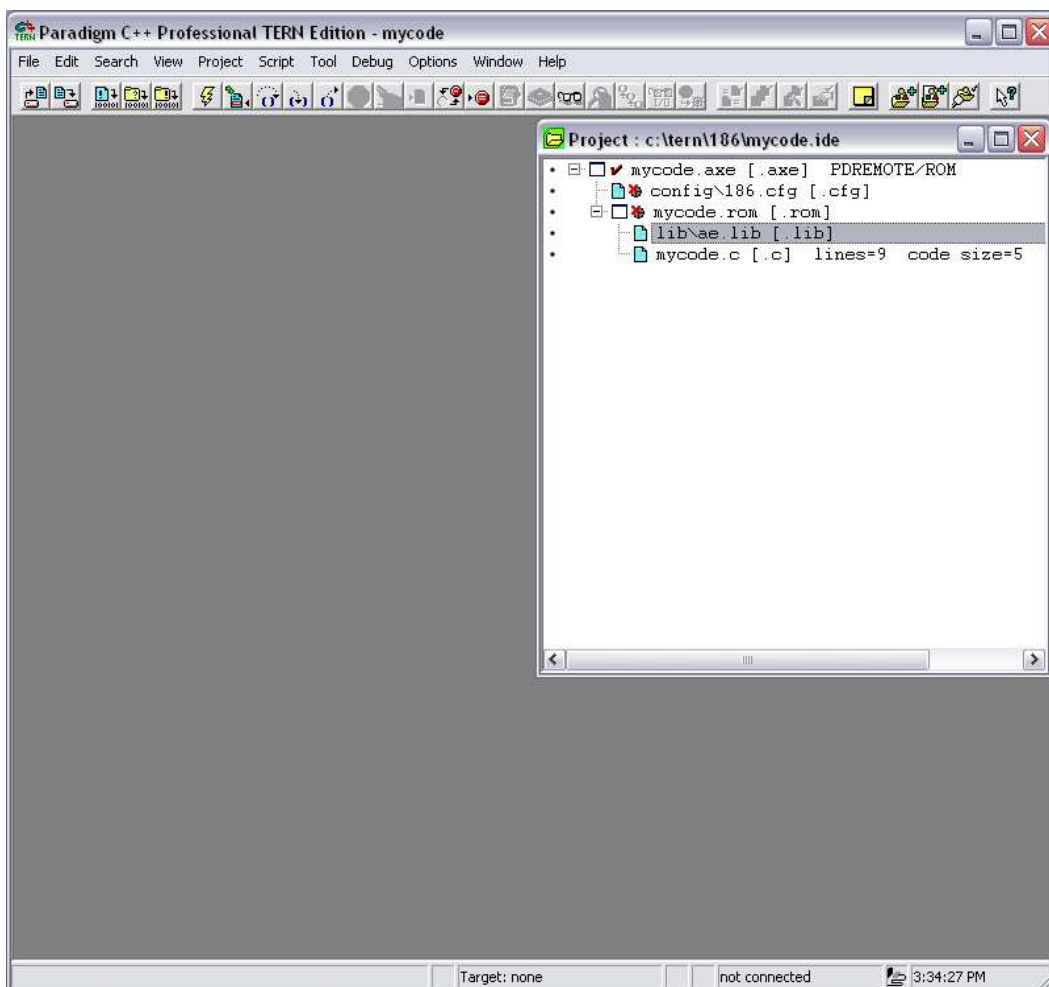
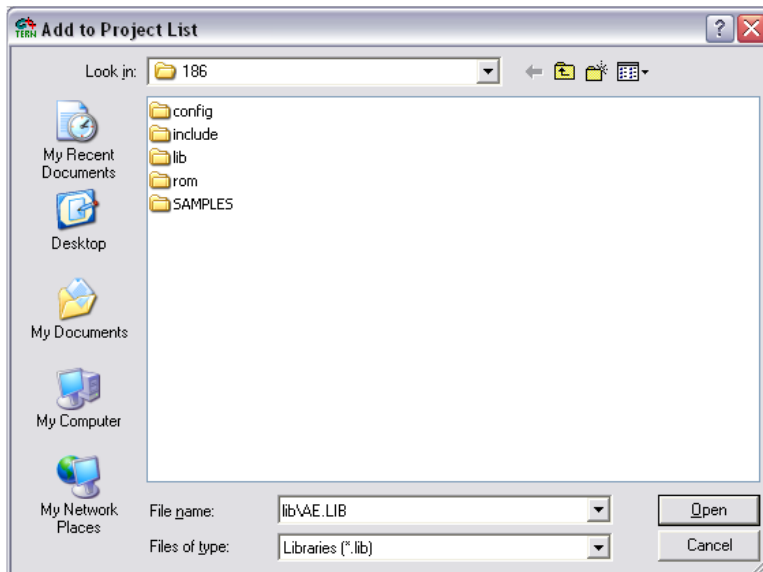


## *How do I add a node like ae.lib?*

Select the parent node of the new child node you are going to add. To add the library “ae.lib” to the node “mycode.rom [.rom]”, right click the “mycode.rom [.rom]” node, and select “Add node” from the popup menu.



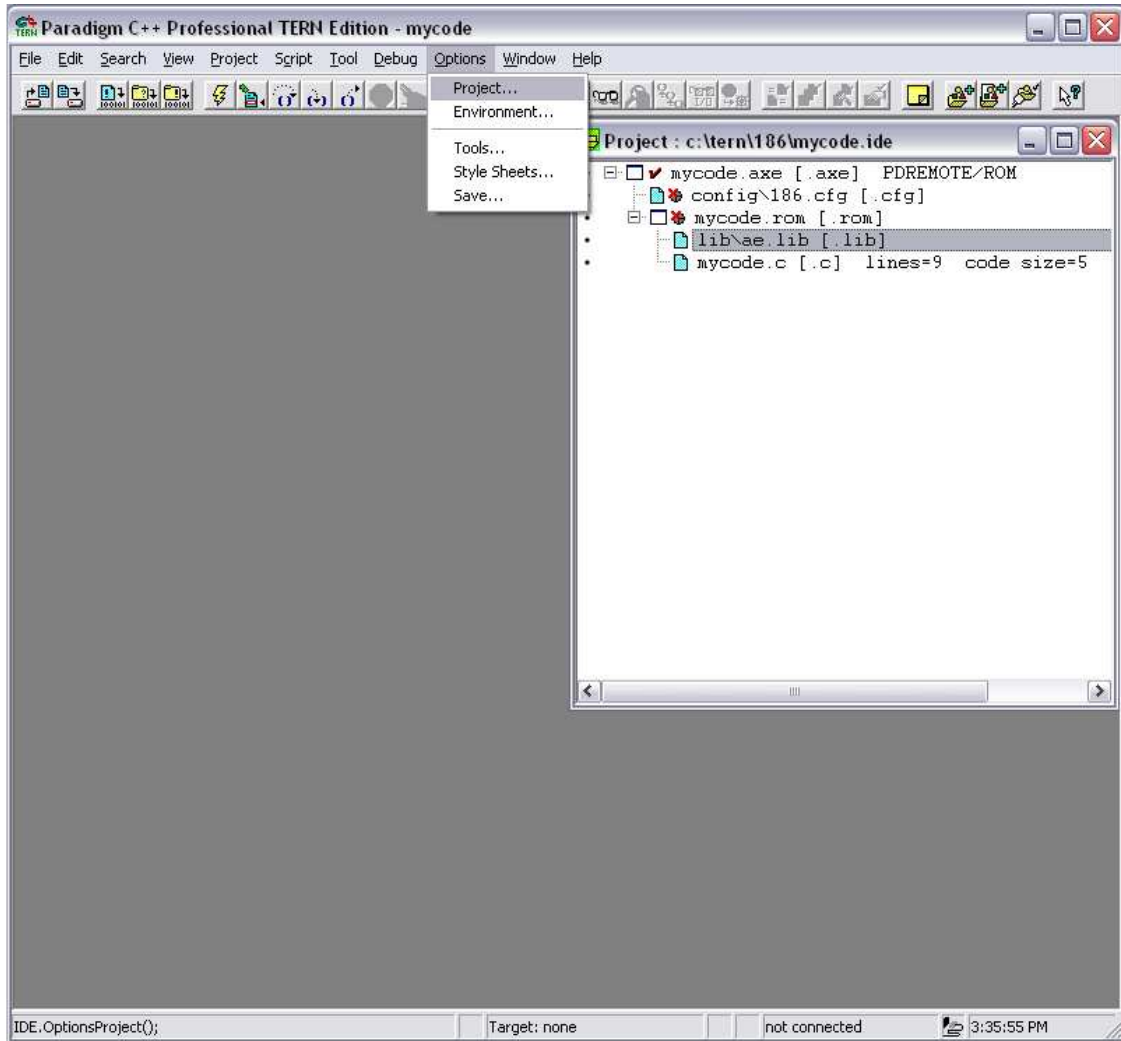
You can type in “lib\ae.lib” or use the mouse to select the library.



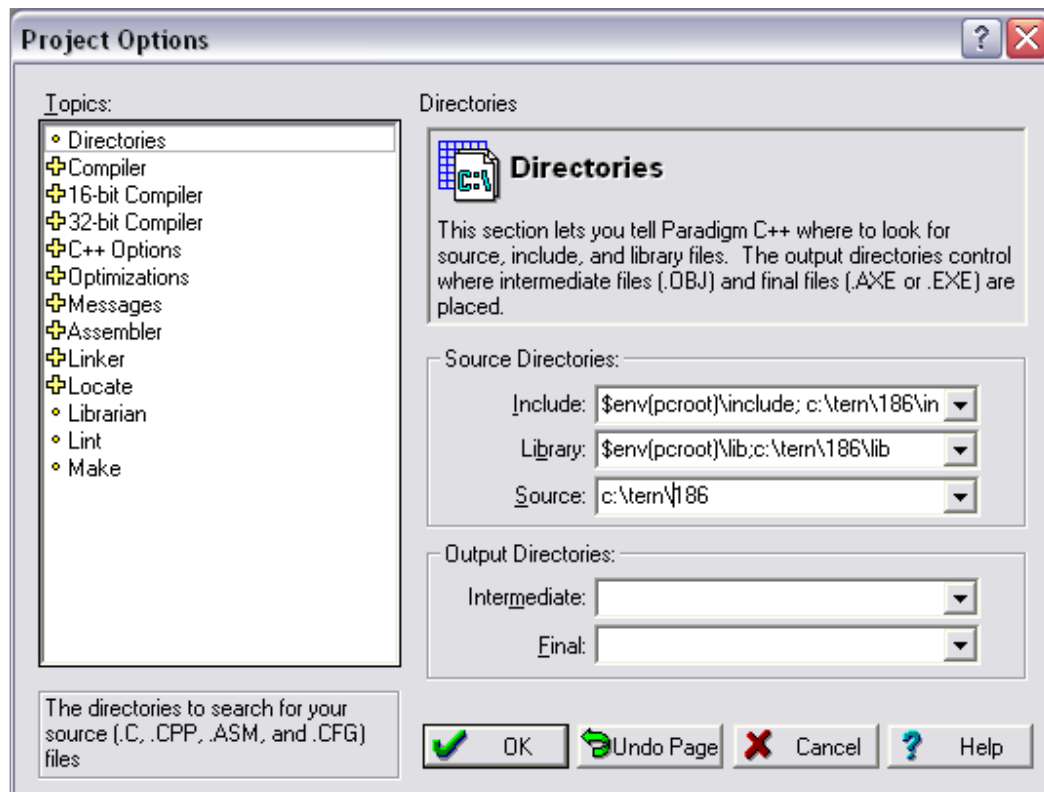


## *How do I setup the Project and Environment?*

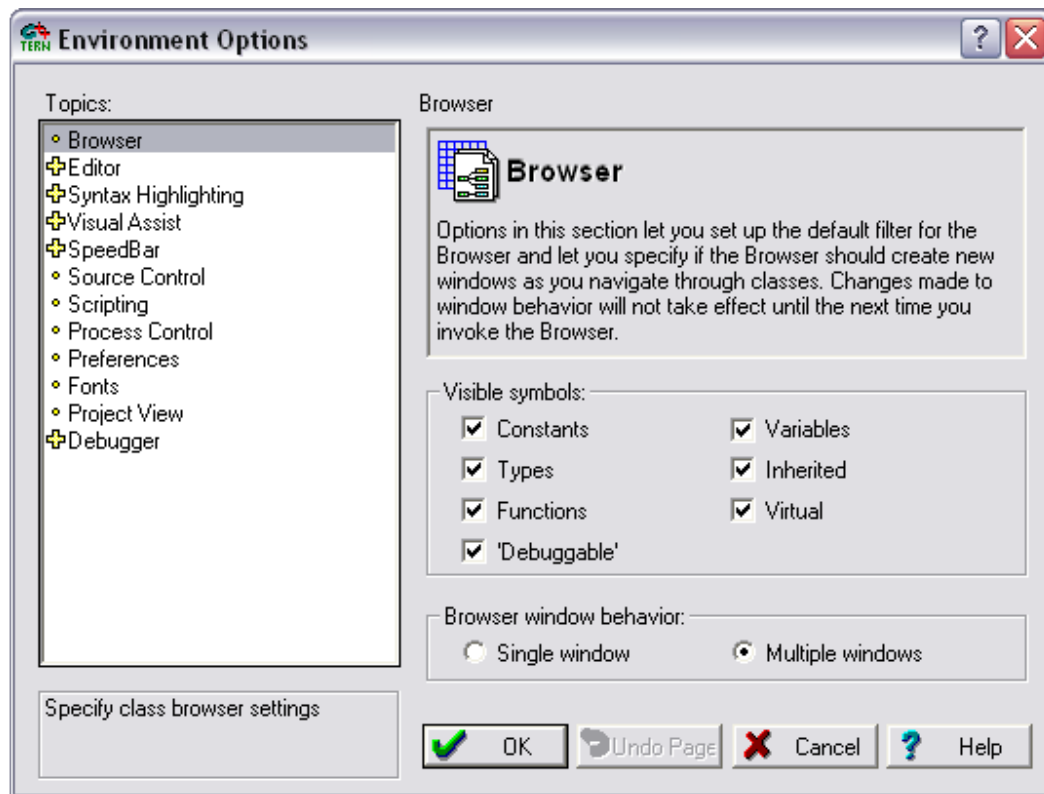
To edit a project's options, select 'Options', then 'Project' from the main menu. To edit the workspace environment, select 'Options', then 'Environment' from the main menu.



The 'Project Options' window allows you to set properties for all of the nodes in your project. Important items to configure are the directories for include, lib and source files. You must add the working directory to Include, Library and Source, as shown below.



Environment Options lets you change how you interface with Paradigm C++ Professional – TERN Edition. These settings allow you to change things like the font of your code or the behavior of the Debugger.



## Chapter 4: Questions & Answers

### 4.1 How is the software organized on my PC?

Paradigm C++ Professional – TERN Edition installs TERN libraries and sample projects on the C: drive of your PC in the C:\TERN directory.

TERN libraries and sample programs are organized by CPU type. Am188/Am186 CPU projects are in the C:\TERN\186 directory, i386EX CPU projects are in the C:\TERN\386 CPU directory, SC520/Am586 CPU projects are in the C:\TERN\586 directory, and V25 CPU projects are in the C:\TERN\V25 directory.

The following directories are created after installing Paradigm C++ Professional – TERN Edition:

```
C:\TERN\  
  
C:\TERN\186  ----- working directory for Am188/6 based controllers  
C:\TERN\186\SAMPLES\ ---- sample programs  
C:\TERN\186\CONFIG\ ---- configuration files  
C:\TERN\186\INCLUDE\ ---- Header files  
C:\TERN\186\LIB\ ----- Libraries for 186 boards  
  
C:\TERN\386  ----- working directory for i386EX based controllers  
C:\TERN\386\SAMPLES\ ---- sample programs  
C:\TERN\386\CONFIG\ ---- configuration files  
C:\TERN\386\INCLUDE\ ---- Header files  
C:\TERN\386\LIB\ ----- Libraries for 386 boards  
  
C:\TERN\586  ----- working directory for 586 based controllers  
C:\TERN\586\SAMPLES\ ---- sample programs  
C:\TERN\586\CONFIG\ ---- configuration files  
C:\TERN\586\INCLUDE\ ---- Header files  
C:\TERN\586\LIB\ ----- Libraries for 386 boards  
  
C:\TERN\V25  ----- working directory for V25 based controllers  
C:\TERN\V25\SAMPLES\ ---- sample programs  
C:\TERN\V25\CONFIG\ ---- configuration files  
C:\TERN\V25\INCLUDE\ ---- Header files  
C:\TERN\V25\LIB\ ----- Libraries for V25 boards
```

### 4.2 Where is the initial working directory?

If you are using V25 based controllers, your working directory is c:\tern\v25.

If you are using 186 based controllers, your working directory is c:\tern\186.

If you are using 386 based controllers, your working directory is c:\tern\386.

If you are using 386 based controllers, your working directory is c:\tern\586.

### 4.3 What is the Project file?

The project file stores the structure and settings of a project. Paradigm C++ Professional – TERN Edition project files have the suffix **PDL**.

## 4.4 Controller Memory mapping

16-bit TERN controllers have a 1 MB memory space. TERN controllers map the lower 512KB (0x00000-0x7ffff) to the read-write SRAM and the upper 512KB to the read-only EPROM/FLASH.

In order to correctly download a program in STEP1 via PC RS232 serial link, the controller must meet the following requirements:

- 1). DEBUG EPROM “AE\_0\_115” for A-Engine, SER0 debug at 115,200 installed in ROM socket.  
See attached DEBUG ROM list

- 2). SRAM installed must be large enough to hold your program.  
For a 32K SRAM, the physical address is 0x00000-0x07fff  
For a 128K SRAM, the physical address is 0x00000-0x01ffff  
For a 512K SRAM, the physical address is 0x00000-0x07ffff

- 3). Your program must be located at a correct starting address to match your SRAM size.  
For example, if you have a 32K SRAM, which maps into 0-0x07fff. Even for a 10K program, if you locate the code into a starting address of 0x08000, it cannot be download nor run because there is no RAM.

- 4). Your program is located by a configuration file,  
c:\tern\186\config\186.cfg, or  
c:\tern\v25\config\v25.cfg, or  
c:\tern\386\config\386.cfg.  
c:\tern\586\config\586.cfg.

- 5). In the “186.cfg” file, you will see these lines:

```
map    0x00000 to 0x00fff as reserved    // interrupt vector table
map    0x01000 to 0x07fff as rdwr        // System RAM area(32 K)
map    0x08000 to 0x1ffff as ronly       // Simulated ROM area(96K)
map    0x20000 to 0xfffff as reserved    // No access
dup    DATA ROMDATA                    // Make a copy of initialized data
class  CODE = 0x0800                    // Assume loading at address 080000H
class  DATA = 0x0100                    // Data at address 01000H
```

The bolded 0x8000 indicates your code will be located at a start address of 0x0800:0000.

- 6). It is very important to know, what size SRAM you have, where is your effective SRAM address, and where you located your program while you are using the “186.cfg” file.

If you have a only 32K SRAM (0-0x7fff), you may modify the “186.cfg” file to select RAM=32.

```
map    0x00000 to 0x00fff as reserved    // interrupt vector table
map    0x01000 to 0x03fff as rdwr        // System RAM area(16 K)
map    0x04000 to 0x07fff as ronly       // Simulated ROM area(16K)
map    0x08000 to 0xfffff as reserved    // No access
dup    DATA ROMDATA                    // Make a copy of initialized data
class  CODE = 0x0400                    // Assume loading at address 04000H
class  DATA = 0x0100                    // Data at address 01000H
```

The bolded 0x4000 indicates your code will be located at a start address of 0x0400:0000.

- 7). In order to run STEP2 stand alone mode, you must know where your code are located during STEP1 downloading. You must correctly set the STEP2 jumping address in your serial EEPROM (EE address 0x10 to 0x13). You may run the “step2.c” program to setup your EE jumping address. You only need to run the “step2.c” once. Please see “step2.c” in c:\tern\186\samples\ae directory.

## 4.5 DEBUG baud rates

Paradigm C++ Professional – TERN Edition supports one of the debug Baud Rate: 19,200, or 38,400, or 57,600, or 115,200. The DEBUG ROM installed in your device determines the DEBUG Baud Rate. For example: DEBUG ROM labeled “AE\_0\_115” installed on a 40 MHz A-Engine, the debug baud rate is 115,200. The same debug rom installed on a 20 MHz A-Engine, the baud rate is 57,600.

## 4.6 What are differences between the Evaluation version and full version?

The Evaluation version supports STEP1 and STEP2 only, not STEP3.

In addition to all the Evaluation version features, the full version provides support that you will need to generate your own application ROM, STEP3.

## 4.7 What is ACTR/ACTF™?

*Operating TERN's 16-bit controllers with ACTR/ACTF™*

With ACTR™/ ACTF™ firmware support, you only need a terminal, such as PC windows terminal, set up to 19200, 8, N, 1, via serial link. By typing text commands, you can operate the controller and exercise C functions immediately. The ACTR/ACTF™ provides an interactive menu and on-line help for you, so you do not have to dig into manuals. ACTR/ACTF™ not only provides you easy access to C functions, but also allows you to download a remote debugger kernel for your C/C++ program development.

ACTR™ is in a 64K EPROM. ACTF™ is a FLASH version of ACTR™. You may use DV kit to generate your application HEX file and download the file into the on-board Flash with ACTF™.

ACTF™ can be used in your final OEM product. As long as the jumper is on every time the power is on or reset, the controller always runs your program. You may download several application programs in different memory locations, remove the jumper, use ACTF™ “Gxxxxx” commands to set up a new CS:IP, and when the jumper is restored, it will run that program at power on or reset

## 4.8 Where is the technical manual?

Complete sets of Technical Manuals, photos, schematics, and help manuals are in the Paradigm C++ Professional – TERN Edition CD. You may use Windows Explore to copy, use Adobe Reader to read and print.

## 4.9 What is the .MAP file and the .LOC file?

After you “Build node” on the .axe [.axe] node, or run the code on the controller it will generate a .map file and a .loc file, in your working directory, c:\tern\186.

The led.map file, as below, will list generated code and data locations.

Start	Stop	Length	Name	Class
00000H	0187DH	0187EH	_TEXT	CODE
01880H	01897H	00018H	_INIT_	INITDATA
01898H	01898H	00000H	_INITEND_	INITDATA
01898H	01898H	00000H	_EXIT_	EXITDATA
01898H	018A7H	00010H	_EXITEND_	EXITDATA
018B0H	018B0H	00000H	_RD	ROMDATA
018B0H	018BFH	00010H	_ERD	ENDROMDATA
018C0H	01A43H	00184H	_DATA	DATA
01A44H	01A44H	00000H	_CVTSEG	DATA
01A44H	01A44H	00000H	_SCNSEG	DATA
01A44H	01CA1H	0025EH	_BSS	BSS
01CA2H	01CA3H	00002H	_BSEND	BSEND
01CB0H	02CAFH	01000H	_STACK	STACK
02CB0H	02CB0H	00000H	_BFD	FAR_DATA
02CB0H	02CBFH	00010H	_EFD	ENDFAR_DATA
02CC0H	02CC0H	00000H	_BRFD	ROMFARDATA
02CC0H	02CCFH	00010H	_ERFD	ENDROMFARDATA

Program entry point at 0000:0000

The led.loc file, as below, will list generated code and data locations, which is the memory location on the target controller.

### Memory Address Map for Program LED

Start	Stop	Length	Segment	Class
001000H	001183H	00184H	_DATA	DATA
001184H	001184H	00000H	_CVTSEG	DATA
001184H	001184H	00000H	_SCNSEG	DATA
001184H	0013E1H	0025EH	_BSS	BSS
0013E2H	0013E3H	00002H	_BSEND	BSEND
0013F0H	0023EFH	01000H	_STACK	STACK
0023F0H	0023F0H	00000H	_BFD	FAR_DATA
0023F0H	0023FFH	00010H	_EFD	ENDFAR_DATA
008000H	00987DH	0187EH	_TEXT	CODE
009880H	009897H	00018H	_INIT_	INITDATA
009898H	009898H	00000H	_INITEND_	INITDATA
009898H	009898H	00000H	_EXIT_	EXITDATA
009898H	0098A7H	00010H	_EXITEND_	EXITDATA
0098B0H	0098B0H	00000H	_RD	ROMDATA
0098B0H	009A33H	00184H	_DATA	ROMDATA
009A34H	009A34H	00000H	_CVTSEG	ROMDATA
009A34H	009A34H	00000H	_SCNSEG	ROMDATA
009A40H	009A4FH	00010H	_ERD	ENDROMDATA
009A50H	009A50H	00000H	_BFD	ROMFARDATA
009A50H	009A50H	00000H	_BRFD	ROMFARDATA
009A50H	009A5FH	00010H	_ERFD	ENDROMFARDATA

Entry point: 0800:0000

Initial stack: 013F:1000

## 4.10 EPROM and FLASH support.

TERN Controllers that have a socket for EPROM support the devices such as 27C256-70 (32K) or 27C512-70 (64K). You can burn an EPROM using an EPROM programmer. You can also use the ACTF kit to generate a downloadable .HEX file, and install an FLASH ACTF chip in the ROM socket. With a terminal serial link at 19,200, you can download your code and program the Flash on-board.

TERN Controllers that have on-board surface mount FLASH have the ACTF software built in. These controllers can be programmed using a terminal to download software over a serial connection.

SEE YOUR HARDWARE MANUAL FOR INSTRUCTIONS ON DOWNLOADING SOFTWARE USING ACTF.

## 4.11 How to access 1 MB memory

Most applications use the Small Memory Model on TERN controllers. Small model applications are smaller, more efficient, more reliable, and faster than other models but are limited to a 64KB code space (read-only) and a 64KB data space (read-write). Many users have successfully developed data acquisition type products that store data well beyond the 64K data space of the small memory model.

The TERN Controllers support memory access by using the pokeb and peekb functions.

```
pokeb(segment, offset, data); //sets a byte at the memory located at segment:offset
peekb(segment, offset); //returns a byte at the memory located at segment:offset
```

Paradigm generates a Location (.LOC) file that details where all of the CODE (read-only) and DATA (read-write) of your application are stored on the device. By examining this file, you can determine what unused memory is available to you.

For example, if the DATA area occupies the lower 64KB area of SRAM (0x00000 to 0x0FFFF) and you are using a 128KB SRAM (0x00000 to 0x1FFFF) then the upper 64KB of memory is still available. By using pokeb and peekb, you can access the upper 64KB of memory without violating the small memory model.

Example: Write and read 4,096 (0x1000) bytes from address 0x10000 to 0x11000

```
int i, j;
for(i=0; i<1000; i++)
{
    pokeb(0x1000, i, i);
    j=peekb(0x1000, i);
}
```



## Appendix A: Debug ROM list for TERN controllers

<i>File Name</i>	<i>Size</i>	<i>Lable Name</i>	<i>Debug</i>	<i>Controllors</i>	<i>Other</i>
TDREM_AE.BIN	32K	TDREM_AE 115,200@40M	TD31	AE/A104/TD40/BBA/SL MD88	188 SER0
TDREM_AC.BIN	32K	TDREM_AC 115,200@40M	TD31	AC	188 SER0
TD384_AE.BIN	32K	TDREM_AE 38,400@40M	TD31	AE/A104/TD40/BBA/SLMD88	188 SER0
TDREM_VE.BIN	32K	TDREM_V25 115,200@16M	TD31	VE/CE/TD/V104	V25 SER0
TDREM_IE.BIN	32K	TDREM_i386EX 115,200@66M	TD31	IE/ID/IEP 8-bit SRAM	386 SER0
TD_ID_16.BIN	32K	TD_ID_16 115,200@66M	TD31	ID 16-bit SRAM	386 SER0
TD_IE_16.BIN	32K	TD_IE_16 115,200@66M	TD31	IEP/IEM 16-bit SRAM	386 SER0
<b>AE_0_115.BIN</b>	32K	AE_0_115 115,200@40M	CPP	AE/A104/TD40/BBA/SLMD88	188 SER0
AE_1_115.BIN	32K	AE_1_115 115,200@40M	CPP	AE/A104/TD40/BBA/SLMD88	188 SER1
<b>AE_0_384.BIN</b>	32K	AE_0_384 38,400@40M	CPP	AE/A104/TD40/BBA/SLMD88	188 SER0
AE_1_384.BIN	32K	AE_1_384 38,400@40M	CPP	AE/A104/TD40/BBA/SLMD88	188 SER0
<b>AC_0_115.BIN</b>	32K	AC_0_115 115,200@40M	CPP	AC	188 SER0
AC_1_115.BIN	32K	AC_1_115 115,200@40M	CPP	AC	188 SER1
AC_0_384.BIN	32K	AC_0_384 38,400@40M	CPP	AC	188 SER0
AC_1_384.BIN	32K	AC_1_384 38,400@40M	CPP	AC	188 SER1
<b>IE8_115.BIN</b>	32K	IE8_0_115 115,200@66M	CPP	IE/ID/IEP 8-bit SRAM	386 SER0
IE8_384.BIN	32K	IE8_0_384 38,400@66M	CPP	IE/ID/IEP 8-bit SRAM	386 SER0
IE8_115D.BIN	32K	IE8_0_115 DCD1 115,200@66M	CPP	IE/ID/IEP 8-bit SRAM	386 SER0 DCD1
IE8_384D.BIN	32K	IE8_0_384 DCD1 38,400@66M	CPP	IE/ID/IEP 8-bit SRAM	386 SER0 DCD1
ID16_115.BIN	32K	ID16_0_115 115,200@66M	CPP	ID 16-bit SRAM	386 SER0
IE16_115.BIN	32K	IE16_0_115 115,200@66M	CPP	IEP/IEM 16-bit SRAM	386 SER0
ID16_384.BIN	32K	ID16_0_384 38,400@66M	CPP	ID 16-bit SRAM	386 SER0
IE16_384.BIN	32K	IE16_0_384 38,400@66M	CPP	IEP/IEM 16-bit SRAM	386 SER0
<b>VE_0_115.BIN</b>	32K	VE_0_115 115,200@16M	CPP	VE/CE/TD/V104	V25 SER0
VE_1_115.BIN	32K	VE_1_115	CPP	VE/CE/TD/V104	V25 SER1

		115,200@16M			
VE_0_384.BIN	32K	VE_0_384 38,400@16M	CPP	VE/CE/TD/V104	V25 SER0
VE_1_384.BIN	32K	VE_1_384 38,400@16M	CPP	VE/CE/TD/V104	V25 SER1