

## Ashling Product Brief APB1 73

# Getting Started with the Keil 8051 Compiler; integrating Ashling's PathFinder-51 Debugger with the Keil $\mu$ Vision IDE

### Contents

1	Introduction .....	1
2	Starting your first project with the Keil $\mu$ Vision IDE for 8051 .....	2
3	Setting-up the Keil $\mu$ Vision IDE for use with PathFinder-51 .....	3
3.1	Set the Keil Compiler and Linker debug options.....	3
3.2	Configure Keil $\mu$ Vision to run the Ashling SymFinder-OMF utility.....	3
3.3	Create a PathFinder Debugger option on the $\mu$ Vision menu .....	4
4	Using the Keil $\mu$ Vision IDE with the Ultra-51 Emulator and PathFinder-51 Debugger .....	4
5	Using the Keil Compiler and Linker in Command-line mode.....	4
6	Troubleshooting Symbols Conversion with SymFinder-OMF.....	5

## 1 Introduction

The Keil 8051 Compiler and Assembler provide a powerful, flexible and reliable toolkit for creating code for all 80C51-architecture microcontrollers. In addition, Keil's  $\mu$ Vision-2 Integrated Development Environment (Fig. 1) provides a friendly, well-structured workbench for defining 8051 projects, editing C and Assembly source files, and controlling the Compiler and Assembler.

This Ashling Product Brief shows, using the Ashling CONTROLR example-program, how to set-up a Project in Keil's  $\mu$ Vision IDE. It also describes how to configure the  $\mu$ Vision IDE for use with Ashling's PathFinder-51 Debugger and Ultra-51 Emulator so that you can rapidly and easily invoke the PathFinder debugger and download your code to the Ultra-51 emulator from the  $\mu$ Vision menu-bar.

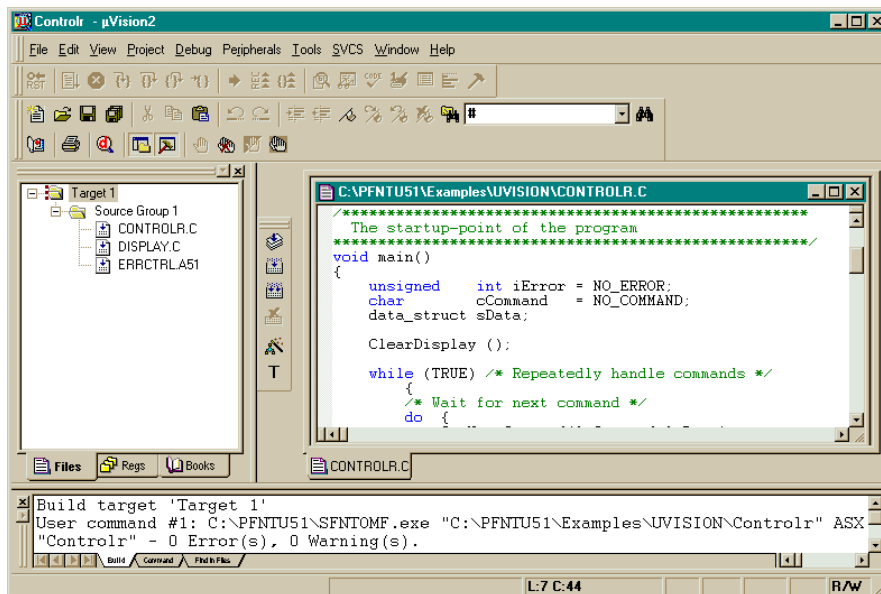


Fig. 1. The Keil  $\mu$ Vision Integrated Development Environment

You can compile C files by either:

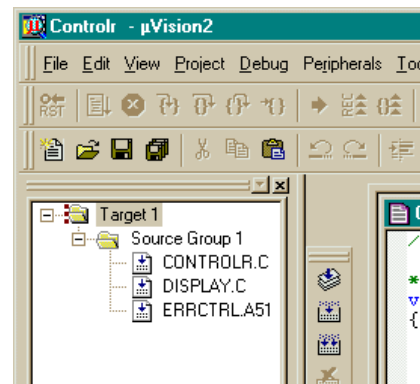
- i. Compiling and linking your files using the Keil  $\mu$ Vision Integrated Development Environment; or
- ii. Running the Keil 8051 C Compiler and Linker from a Batch-file or Make-file.

## 2 Starting your first project with the Keil $\mu$ Vision IDE for 8051

In this “Getting-Started” example, we’ll set-up and use Keil’s  $\mu$ Vision IDE to compile and link the CONTROLR demo-program that is supplied with Ashling’s PathFinder-51 debugger in the directory C:\PFNTU51\EXAMPLES\KEILDEMO, using the following steps:

1. Use Windows Explorer to create a new directory for this  $\mu$ Vision example:  
C:\PFNTU51\EXAMPLES\UVISION
2. Use Windows Explorer to Copy the following four Source Files:  
CONTROLR.C            GLOBAL.H  
DISPLAY.C            ERRCTRL.ASM  
from  
C:\PFNTU51\EXAMPLES\KEILDEMO to C:\PFNTU51\EXAMPLES\UVISION
3. Use Windows Explorer to re-name the 8051 Assembly file  
C:\PFNTU51\EXAMPLES\UVISION\ERRCTRL.ASM to ERRCTRL.A51 because  $\mu$ Vision uses a default .A51 suffix for Assembly files.
4. Run Keil  $\mu$ Vision-2. Select **Project->New project....** In the file-dialog, go to the directory C:\PFNTU51\EXAMPLES\UVISION that you have created, and enter a name for this new project, **Controlr** ( $\mu$ Vision will add a project-file suffix and create **Controlr.uv2**)
5.  $\mu$ Vision will next ask you to **Select a Device for Target ‘Target 1’**  
Select **Philips**, then **80C32** for this example.
6. Next, we’ll add the two C Source Files and the Assembly File to the project.  
Select **Project->Targets, Groups, Files...->Groups/Add Files** tab.  
In the **Available Groups** section, click on **Source Group 1**  
then press **Add Files to Group...**  
In the **Add Files to Group ‘Source Group 1’** file-dialog, go to the C:\PFNTU51\EXAMPLES\UVISION directory and add files (press **Add** for each file):  
CONTROLR.C            (C Source file)  
DISPLAY.C            (C Source file)  
ERRCTRL.A51            (Asm Source file)
7. The Files tab in  $\mu$ Vision’s Project window now lists the three Source files; see Fig. 2.

Fig. 2.  $\mu$ Vision’s Project window, Files tab shows all of the source-files in your CONTROLR project



8. The CONTROLR program uses the 8051 Large memory-model (variables are stored in Xdata memory, starting at 0x1000). Configure this memory-model using **Project->Options**

for Target {current target}. On the Target tab, select **Memory Model: Large: variables in XDATA** and set **Off-chip Xdata memory** to **Ram Start: 0x1000**, then press **OK**

9. At this stage, you have given  $\mu$ Vision all of the information that it needs to edit, compile and link the CONTROLR program. You can select **Project->Build target** to compile, assemble and link all of the files in the CONTROLR project.

### 3 Setting-up the Keil $\mu$ Vision IDE for use with PathFinder-51

Setting-up the Keil's  $\mu$ Vision IDE for debugging with Ashling's Ultra-51 Emulator and PathFinder-51 Debugger requires three simple steps:

- i. Select the appropriate Compiler and Linker commands and options to ensure that your Keil output file contains Debug information;
- ii. Configure  $\mu$ Vision so that it automatically runs the Ashling "SymFinder-OMF" Symbol Processing Utility program after every link, to prepare your 8051 executable file for debugging using PathFinder-51; and
- iii. Create a "PathFinder debugger" menu-option in  $\mu$ Vision so that you can invoke PathFinder-51 and load your current 8051 program for debugging.

#### 3.1 Set the Keil Compiler and Linker debug options

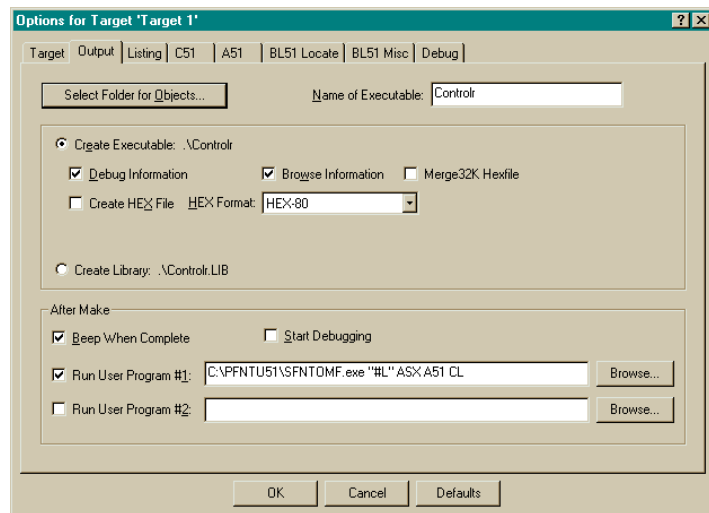
By default,  $\mu$ Vision sets-up the Keil Compiler with the correct options and commands for debugging. To confirm your current settings, open your Keil project in  $\mu$ Vision and go to the  $\mu$ Vision **Project->Options for Target {current target}** menu:

- a. On the **Output** tab, ensure that **Create Executable** and **Debug Information** are selected.
- b. On the **Listing** tab, ensure that **Linker Listing** and **Memory Map** are selected; all other Listing selections are optional.
- c. On the **C51** tab, ensure that the default **DEBUG OBJECTTEXTEND** compiler controls are specified; all other Compiler controls and selections are optional.
- d. On the **A51** tab, ensure that the default **DEBUG** assembler control is specified; all other Assembler controls and selections are optional.

#### 3.2 Configure Keil $\mu$ Vision to run the Ashling SymFinder-OMF utility

- a. Open your Keil project in  $\mu$ Vision, from **Project->Open Project**
- b. Go to the  $\mu$ Vision **Project->Options for Target {current target}** menu. On the **Output** tab, tick the **Run user program #1** box (see Fig. 3) and enter:  
**C:\PFNTU51\SFNTOMF.exe "#L" ASX A51 CL**

*Fig. 3. Use the **Output** tab on  $\mu$ Vision's **Project->Options for Target** dialogs to run the Ashling SymFinder SFNTOMF utility after each  $\mu$ Vision Make*



This tells  $\mu$ Vision to run the Ashling **SFNTOMF.EXE** (SymFinder-OMF) symbol-processing utility after every Link or (re)Build (Enter the correct path for SFNTOMF.EXE if you didn't install it to the default path).

- "#L" tells  $\mu$ Vision to pass the name of the current linker Output file as a parameter to SymFinder.
- ASX A51** tells SymFinder to use ".A51" as the default Suffix for 8051 Assembly files (Keil assembly files usually use this suffix).
- CL** tells SymFinder to Close after SymFinder processing is complete, if SymFinder encounters no errors.

### 3.3 Create a PathFinder Debugger option on the $\mu$ Vision menu

Open your Keil project in  $\mu$ Vision and go to the **Tools->Customize Tools Menu...** option. In the **Customize Tools Menu** dialog, press  for a New tools-menu item.

- a. In **Menu Content**, enter **&PathFinder debugger**  
(this adds a new PathFinder debugger option to the  $\mu$ Vision Tools menu)
- b. In the **Command** field enter: **C:\PFNTU51\PFPU51.EXE**  
or enter the correct path for the PathFinder Debugger program if you didn't install it in the default path.
- c. In the **Arguments** field enter: **/LD "%L.CSO"**  
(this tells  $\mu$ Vision to invoke PFPU51.EXE with a command to Load a file; the filename is the current  $\mu$ Vision Output file with a ".CSO" suffix. This is the file that will be created by the Ashling SFNTOMF.EXE SymFinder utility).

## 4 Using the Keil $\mu$ Vision IDE with the Ultra-51 Emulator and PathFinder-51 Debugger

Once you have set-up  $\mu$ Vision, using it with PathFinder and the Ultra-51 Emulator is easy:

- After you have created or modified a C or Assembler file, (re)build your Project in  $\mu$ Vision by pressing **Project->Build target** or **Project->Rebuild all target files**. This will compile and link your (updated) files and will run the Ashling SymFinder utility.
- After a successful (re)build in  $\mu$ Vision, you can press **Tools->PathFinder debugger** to run PathFinder and to load your current 8051 program file.
- If you identify a bug in your code using PathFinder and you want to correct it in  $\mu$ Vision, first select **File->Close** in PathFinder's menu; this frees the files from control by PathFinder, allowing a  $\mu$ Vision rebuild.

## 5 Using the Keil Compiler and Linker in Command-line mode

You can also run the Keil Compiler and Linker, and the Ashling SFNTOMF.EXE (SymFinder-OMF) symbol-processing utility, using a Batch-file or within a Make-file. Full details on Batch or DOS commands to run the Compiler, Linker and SymFinder-OMF are in Chapter 11O of the PathFinder Debugger's User manual; see Section 11O.3.1, "A Keil Getting-Started Example".

In summary, a Batch-File or Make-File should include commands to:

1. Compile each C module with the Keil Compiler using:  
C51 {Cfile1}.C OBJECTTEXTEND DEBUG  
C51 {Cfile2}.C OBJECTTEXTEND DEBUG  
C51 {Cfile3}.C OBJECTTEXTEND DEBUG

where the default `OBJECTTEXTEND DEBUG` switches enable Debug information.

2. If your program includes Assembly modules, assemble them with the Keil assembler using:

```
A51 {ASMfile1}.A51 DEBUG
A51 {ASMfile2}.A51 DEBUG
```

3. Link all modules using the Keil Linker, with the command:

```
L51 {Cfile1}.OBJ, {Cfile2}.OBJ, {Cfile3}.OBJ, {ASMfile1}.OBJ, {ASMfile2}.OBJ
```

The Keil Linker, by default, produces an Absolute Output File `{Cfile1}` with the same name as the first Module, without a suffix. Your Linker switches must not alter the default `MAP SYMBOLS PUBLICS` and `LINES` switches.

4. Next, convert the Keil-format Absolute Output File `{Cfile1}` to Ashling format, using the Ashling "SymFinder-OMF" `SFNTOMF.EXE` utility (it's installed in the Ashling program group), as follows:

From DOS, run

```
SFNTOMF {Cfile1}.ABS ASX A51
```

..or from Windows, run `SFNTOMF`, select your Keil-absolute file `{Cfile1}` in the File Name field, and enter `ASX A51` in the "Optional Controls" field.

`SFNTOMF` will create a file `{Cfile1}.CSO` in Ashling C Symbolic Object format.

(The `ASX A51` control tells SymFinder to use ".A51" as the default Suffix for 8051 Assembly files; Keil assembly files usually use this suffix).

5. Load the Ashling PathFinder Debugger, select `File->Load` and choose `{Cfile1}.CSO`

## 6 Troubleshooting Symbols Conversion with SymFinder-OMF

The Ashling `SFNTOMF.EXE` SymFinder-OMF Symbol Processing Utility (supplied with the PathFinder-51 Debugger) searches through the Keil C Absolute Object File for references to your Source files.

If SymFinder-OMF *cannot* locate your C (or Assembler) source files, the utility will show progress messages that guide you towards solving the problem. Chapter 11O in the PathFinder-51 User Manual lists the various SymFinder Controls that you can use to ensure that SymFinder locates all of the necessary files for source debugging, if you are using non-default paths to those files.

**Note** that, to make sure that the SymFinder progress messages remain on the screen, you should *not* use the `CL` control (Close SymFinder if no errors) mentioned in section 3.2. You can re-insert the `CL` control when SymFinder is running correctly.

Fig. 4 shows the screen messages after successful SymFinder processing, showing:

- i. The files, paths and file-suffixes where SymFinder expects to find the Keil Absolute Object File, the C and Assembler Source Files, and the Keil Linker Map File.
- ii. The number of C and Assembler source files located by SymFinder, and their full File-names.
- iii. The total number of Line-number symbols (statement-symbols) located by SymFinder.

```

SymFinder-OMF for NT
SymFinder-OMF, ASLDB, Keil-C and Intel PL/M Source-Debug Converter.
Version 5.0.4, 11-Jul-2002, (c) Ashling Microsystems Ltd 2002.
  Processing controls...
      Input file : C:\PFNTU51\EXAMPLES\UVISION\CONTROLR
      Output file 1 : C:\PFNTU51\EXAMPLES\UVISION\CONTROLR.CSO
      Output file 2 : C:\PFNTU51\EXAMPLES\UVISION\CONTROLR.BUF
      C/ASM/LST path : C:\PFNTU51\EXAMPLES\UVISION\
      Map file : C:\PFNTU51\EXAMPLES\UVISION\CONTROLR.M51
      Output format : Ashling C/ASLDB Symbolic Object.
      C source-file suffix : .C
      PL/M listfile suffix : .LST
      ASM -file suffix : .A51

Windows reports 5288KB of available physical memory.
Using 5288KB of memory for symbol processing.
Processing Map file C:\PFNTU51\EXAMPLES\UVISION\CONTROLR.M51 ...
2 Keil-C-51 source files are available.
1 Assembly Source file is available.
Processing source-file C:\PFNTU51\EXAMPLES\UVISION\CONTROLR.C
Processing source-file C:\PFNTU51\EXAMPLES\UVISION\DISPLAY.C
Processing source-file C:\PFNTU51\EXAMPLES\UVISION\ERRCTRL.A51
Output file contains 46 variable/label symbols
      and 71 statement symbols (line-numbers)
      and 12 user type-definitions.
Current time:12:57:07  Processing time:0 seconds.

```

*Fig. 4. SymFinder-OMF output display; no errors*

Fig. 5 shows SymFinder screen messages after *unsuccessful* processing, showing that:

- iv. SymFinder cannot find the Keil Link Map (.M51) file. This might be because your Linker command file includes the `NOMAP` control; or because your Mapfile isn't located in the same path as the Keil Absolute File (in which case you can use SymFinder's `MAPFILE` or `MA` control to specify a non-default Mapfile name).
- v. Because it cannot locate the Mapfile, SymFinder reports that it cannot locate any C or Assembler source files; and it further reports the Output file contains no Line-number symbols (statement-symbols).

```

SymFinder-OMF for NT
SymFinder-OMF, ASLDB, Keil-C and Intel PL/M Source-Debug Converter.
Version 5.0.4, 11-Jul-2002, (c) Ashling Microsystems Ltd 2002.
  Processing controls...
      Input file : C:\PFNTU51\EXAMPLES\UVISION\CONTROLR
      Output file 1 : C:\PFNTU51\EXAMPLES\UVISION\CONTROLR.CSO
      Output file 2 : C:\PFNTU51\EXAMPLES\UVISION\CONTROLR.BUF
      C/ASM/LST path : C:\PFNTU51\EXAMPLES\UVISION\
      Map file : C:\PFNTU51\EXAMPLES\UVISION\CONTROLR.M51
      Output format : Ashling C/ASLDB Symbolic Object.
      C source-file suffix : .C
      PL/M listfile suffix : .LST
      ASM-file suffix : .A51

Windows reports 5004KB of available physical memory.
Using 5004KB of memory for symbol processing.
Warning : Missing Map file "C:\PFNTU51\EXAMPLES\UVISION\CONTROLR.M51"...
There will be no Source Level Debug information in the output file.
No source files are available.
Output file contains 46 variable/label symbols
      and no statement symbols (line-numbers)
      and 12 user type-definitions.
Current time:12:58:03  Processing time:0 seconds.

```

*Fig. 5. SymFinder-OMF output display; errors encountered*

After you have run SymFinder without errors, load the .CSO file into PathFinder from PathFinder's File->Load menu; then open the Code Browse window from the View menu.

The Code Browse window gives you a "bird's eye view" of all of the C source and Assembler source modules (files) in your program. It confirms that source-debugging information is available for each module, shows the Functions and Line-numbers within each source-module, and allows you to easily "jump" from one module to another, while PathFinder updates the Source and Disassembly windows.

Doc: APB173-V1U-Keil51GettingStarted

**Ashling Microsystems Ltd. is Certified to EN ISO 9001, NSAI Registration No. M619.**

Ashling Microsystems Inc.  
1270 Oakmead Parkway,  
Suite 208  
Sunnyvale, CA 94085  
Tel: (408) 732 6490  
Fax: (408) 732 6497  
Email: [sales.usa@ashling.com](mailto:sales.usa@ashling.com)

Ashling Microsystèmes  
2, rue Alexis de Tocqueville  
Parc de Haute Technologie  
92183 Antony Cedex, France  
Tel: 01.46.66.27.50  
Fax: 01.46.74.99.88  
[sales.fr@ashling.com](mailto:sales.fr@ashling.com)

Ashling Microsystems Ltd  
Intec 2, Wade Road  
Basingstoke  
Hants. RG24 8NE, U.K.  
Tel: (01256) 811998  
Fax: (01256) 811761  
[sales.uk@ashling.com](mailto:sales.uk@ashling.com)

Ashling Microsystems Ltd  
National Technological Park  
Limerick  
Ireland  
Tel: +353-61-334466  
Fax: +353-61-334477  
[sales.ie@ashling.com](mailto:sales.ie@ashling.com)

*Ashling Microsystems Ltd reserves the right to alter product specifications at any time and without notice*

**Distributors in** Austria, Belgium, China, Finland, Germany, Hong Kong, India, Israel, Italy, Korea, Malaysia, the Netherlands, Norway, Singapore, Spain, Sweden, Switzerland, Taiwan and Turkey.

**[www.ashling.com](http://www.ashling.com)**

