Getting Started with C for the FPSLIC™ Family Using the IAR Compiler

Features

- **How to Open a New Project**
- **Description of Option Settings**
- **Linker Command File Examples**
- **Writing and Compiling the C Code**
- **How to Load the Debug and Executable Hex Files into the System Designer™ Software**

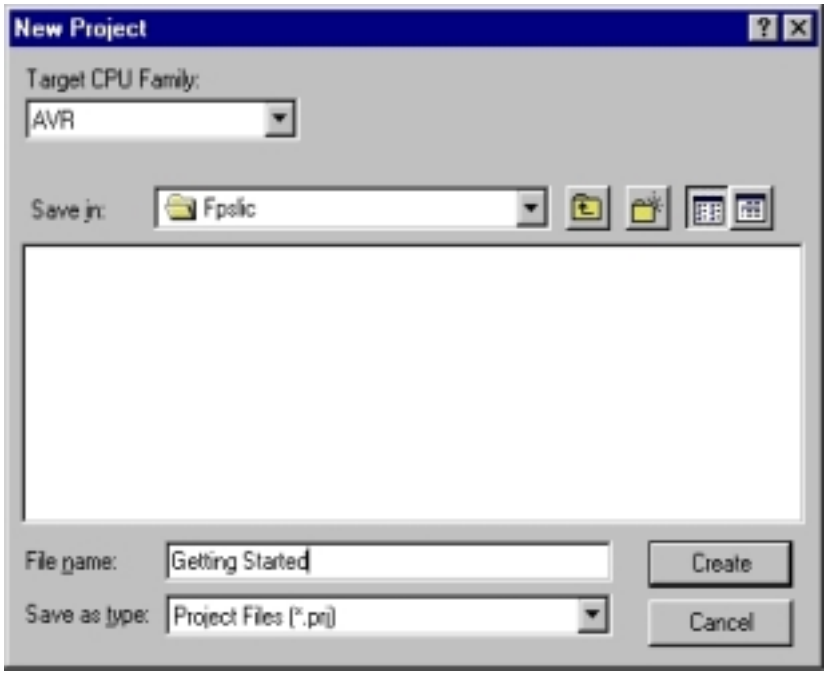
Introduction

The purpose of this application note is to guide new users through the initial settings of the IAR Embedded Workbench™ for AVR® and compile a simple C program. This application note shows how to set up the compiler to generate a UBROF debug file and an executable hex file. The example described in this application note is written for the AT94K using the FPSLIC Starter Kit.

Creating a New Project

Open the IAR Embedded Workbench for AVR and go to the “File” menu and select “New” and then “Project”. The dialog box shown in Figure 1 appears. In this dialog box, first make a folder “C:\FPSLIC” and then type “Getting Started” in the “File name” window. This project should be created in the “C:\FPSLIC” folder.

Figure 1. Create the Project File



FPSLIC™

10K - 40K Gates of AT40K FPGA with 8-bit AVR® Microcontroller and 36K Bytes of SRAM

Application Note

Rev. 1977A–11/00

Settings in “Project > Options”

Before any code can be compiled and linked, the options for the compiler and linker must be set up correctly. By default, it is possible to select two different targets in the “Project” window. The two selections are target “Release” and target “Debug”. The debug target is normally used when running the code in a simulation environment, while the release target is normally used when producing code that can be executed on a real device. The settings done in the “Project > Options” menu are individual for both targets. Thus, it is necessary to set all options twice when using both targets, as in this application note. The main difference between the two targets is the format of the output file.

It is also possible to add more targets whose options can be customized to a specific FPSLIC (simulated or the real device). Common and different source files may be included in the different targets. A folder will be created for each target when linked for the first time.

To set up a common source folder, simply navigate to “Project > New Group” and enter the name “Common Sources” in “Group Name”. By default, both targets are selected, so the group will be added to both targets. After clicking “OK”, the group will be displayed in the “Project” window.

In this application note, the goal is to make a file that can be imported into System Designer for both simulation and execution on an AT94K device. To do this, both debug and release targets will be used. Select the “Debug” target in the “Getting Started.prj” window, as shown in Figure 2. Then select the “Project > Options” menu. A window similar to the one in Figure 3 should pop up.

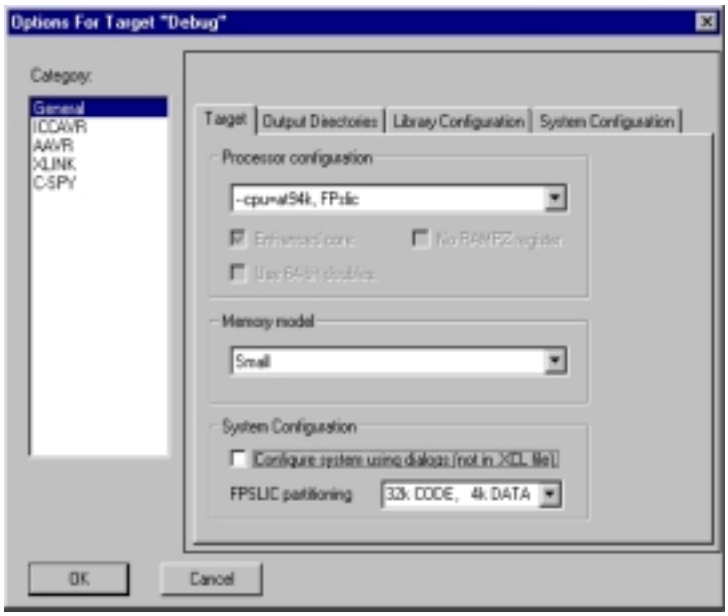
Figure 2. Selecting Target Debug

General Settings

In the “General” category in the “Options” dialog box, the type of processor used is selected. Under “Processor configuration” select “—cpu=at94k, FPslic” and choose “Small” for the “Memory model”. “Memory model: Small” uses a 2-byte data pointer, thus allowing up to 64 KB of data. Under “System Configuration” uncheck the “Configure system using dialogs” box and select the desired “FPSLIC partitioning”. For this example, select “32k CODE, 4k DATA”. It is necessary to change one other option. Select the “General > Library Configuration” tab and check the “Enable bit definitions in I/O-Include files” box.

In our example, the settings shown in Figure 3 should be used.

Figure 3. General Options Dialog



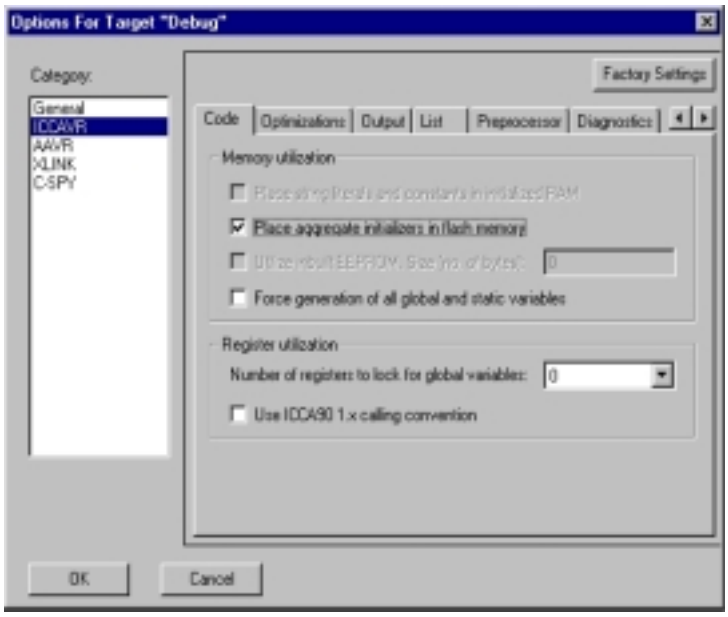
ICCAVR Settings

To get the dialog options for the specific settings of the compiler, click on the “ICCAVR” line in the “Category” list.

The compiler may be optimized for code size or execution speed. The type and level of optimization may be set in the “Optimizations” group in Figure 4. Only one type of optimization may be specified for a single target. Note that if a high level of optimization is used, the user may not be able to debug the code. The code will be fully debuggable with optimization level of “Medium” or lower.

On the “List” tab, the user is able to determine whether a C listing is generated, and the information included in this listing. The “Assembler mnemonics” option will, if checked, cause the compiler to include the generated assembly lines in the listing. It is extremely important that “Assembler file” is not checked, as this will not allow the proper simulation of the FPSLIC MCU in the System Designer software.

Figure 4. ICCAVR Option Settings



AAVR Settings

In the AAVR settings, the options for the assembler can be changed. Since this application note does not contain any parts written in assembly, the default settings can be left unchanged.

XLINK Settings

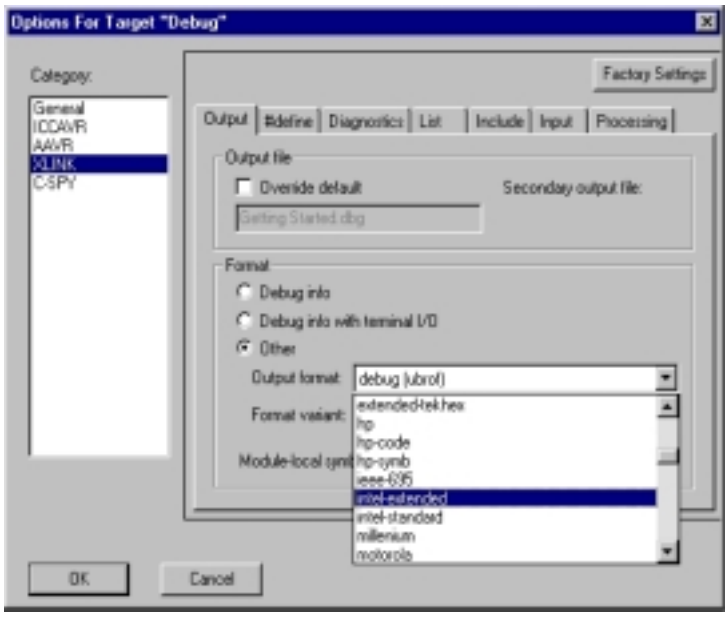
The linker settings give the linker instructions on how to link together the object codes from the different Compiler, Assembler and Library modules.

The first thing that needs to be selected is the format of the output file the linker is to create. In this application note, the intention is to generate both a “debug[ubrof]” and an “Intel Extended HEX” file, which are recognized by System Designer and the FPSLIC Starter Kit, respectively.

Selecting the “Output” tab of the “XLINK” options and clicking the “Other” button in the format section does this. Select “intel-extended” from the “Output format” pull-down menu, as shown in Figure 5, if your target is set to “Release”. Otherwise, select “debug[ubrof]” from the “Output format” pull-down menu and select “Old UBROF version” from the “Format variant” pull-down menu if your target is set to “Debug”.

In the “Output file” group it is possible to rename the output file. The default name is the same as the project name.

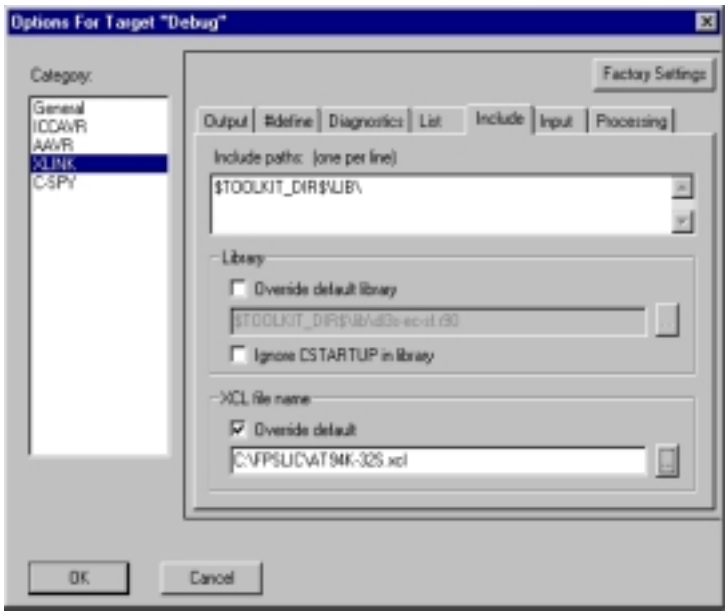
Figure 5. Selecting Output Format



The other thing that has to be changed is the “Linker Command File” used. To change this, click on the “Include” tab and under the “XCL file name” bar, check “Override default”, as shown in Figure 6. Then click the “…” button and navigate to the “AT94K-32S.xcl” file attached to this application note. Here it is assumed that the file is stored in the “C:\FPSLIC” directory.

The main purpose of the “Linker Command File” is to determine the code and data segments, which is done in the –Z command. Note that the size of the Data Stack and Return Stack is specified explicitly and may be changed according to a specific project. The “Linker Command File” will probably need to be edited for each project. “The Linker Command Files” attached to this application note must be considered as a starting point only. Please see the application note “Linker Command Files for the IAR ICCAVR Compiler for the FPSLIC Family” for how to modify the “Linker Command File” to fit the specific project.

Figure 6. Selecting the XCL File



Writing the Source File

When the “Project” options are properly configured, the next step is to write the source code. This application note uses a simple program that increments the PORTD to which the eight LEDs are attached. An 8-bit timer is used to generate a delay between incrementation, making it possible to see the LEDs flashing.

To open a new source file, select “File > New” and then select “Source/Text”. In the new window that appears, type in the text below and save it as “FPSLIC.C” by selecting “Save As” in the “File” menu. Make sure to save the file in the “C:\FPSLIC” directory.

Program Listing for AT94K

```
#include <ioat94k.h>


/* Function Prototypes */
void initialization(void);
void delay(void);


/* Function Definitions */
void initialization(void)
{
        DDRD  = 0xFF;           /* Set PORTD as Output */
        TCCR0 = 0x05;           /* Count SysClock/1024 */
}

void delay(void)
{
        while(!(TIFR & 0x02));  /* Wait for TC0 Overflow Flag to Set */
        TIFR  = 0x02;           /* Clearing Overflow Flag */
}


/* Main Program */
void main(void)
{
        initialization();       /* Initialize Peripherals */

        for(;;)                 /* Infinite Loop */
        {
                PORTD++;        /* Increment PORTD */
                delay();        /* Short delay */
        }
}
```

The program is divided in three parts: initialization, delay and the main program. In the initialization part, PORTD is set as an output and Timer/Counter0 starts to count the system clock divided by 1024.

In the delay subroutine, the controller waits for the Timer/Counter0 overflow flag to be set, then clears the flag and exits.

In the main program, the content in PORTD is incremented and a delay is called to make the change on PORTD visible.

Including the Source File in the Project

When the source code is written, it has to be included in the project. Selecting “Files” from the “Project” menu does this. The dialog box shown in Figure 7 appears. Navigate to the “C:\FPSLIC” folder, select the file “FPSLIC.C” by clicking on it and select “Add”. Click “Done” to exit the dialog box.

Figure 7. Selecting Source Files

Compiling the Code

To compile the code, select “Project > Make” or press the “F9” key. If everything is done correctly, the code compiles and links with no errors and the debug[ubrof] code is placed in the file “C:\FPSLIC\DEBUG\EXE\GETTING STARTED.DBG” if the target is set to “Debug”. Alternatively, if the target is set to “Release”, the executable hex code is placed in the file “C:\FPSLIC\RELEASE\EXE\GETTING STARTED.A90”.

Loading the File into System Designer

To simulate and run the code, the file must be loaded into System Designer. This application note describes how to load it into System Designer and the FPSLIC Starter Kit.

When doing any sort of co-simulation, System Designer requires some sort of object file, depending on which compiler is being used, for the FPSLIC MCU simulation. The object file generated by the IAR Systems Embedded Workbench for AVR Compiler is the UBROF file, generated when the target is set to “Debug”. Whenever it is necessary to load the object file into AVR Studio™, select “C:\FPSLIC\DEBUG\EXE\GETTING STARTED.DBG”.

When generating the combined bitstream in System Designer and you are prompted for the software file, select the executable hex file, which is located at “C:\FPSLIC\RELEASE\EXE\GETTING STARTED.A90”.

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Atmel Smart Card ICs
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-803-000
FAX (44) 1355-242-743

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Atmel FPSLIC Hotline
1-(408) 436-4119

Atmel FPSLIC e-mail
fpslic@atmel.com

FAQ
Available from Website

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

1977A–11/00/xM