Nuvoton 8051 IAR ICE Driver User Manual

Nuvoton is providing this document only for reference purposes of Nuvoton 8051 microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

Table of Contents

List of Figures

1 Overview

1.1 Introduction

The Nuvoton 8051 IAR Driver allows the IAR Embedded Workbench to communicate with Nuvoton on-chip debug logic.
In-system flash memory programming integrated into the driver allows the user to rapidly update target code. The IAR Embedded Workbench can be used to start and stop program execution, set breakpoints, check variables, inspect and modify memory contents, and single-step through programs to run your actual target hardware.
This document describes how to install and use 8051 IAR Driver with programs written using IAR's compiling and flash tools.

1.2 Features

The Nu-Link driver supports the following features. Some functions are triggered by IAR. The usage of these functions can be found in the IAR User Guide.

- Erase/program/verify Nuvoton chips. (via flash algorithm of Nu-Link IAR driver)
- Easy registers access of Nuvoton chips. (via the .ddf file of Nu-Link IAR driver)

1.3 Supported Devices

Open hyperlink to see supported 8051 devices: Link (search for 8051 series).

2 Installing Nu-Link IAR Driver

2.1 System Requirements

- **Software:** IAR Embedded Workbench for 8051
- **Hardware:** Nu-Link ICE adapter

2.2 Installation

Double click Nu-Link_IAR_Driver.exe, the following directories and files can be found after package is installed successfully:

- **.\Samples:** The sample project that uses Nu-Link driver for IAR.
- **.\ Nu-Link_IAR_51.dll:** The driver DLL.

3 Example – Create and Debug a Project

3.1 Start a New Project

This section describes how to start a new project based on 8051 series chips. The fast and easy way to start a new project is open an existing IAR project. To make sure the user knows about all the steps to create an IAR project, this section will start with an empty project.

1. Open IAR Embedded Workbench, and click "**File**" → "**New**" → "**Workspace**".

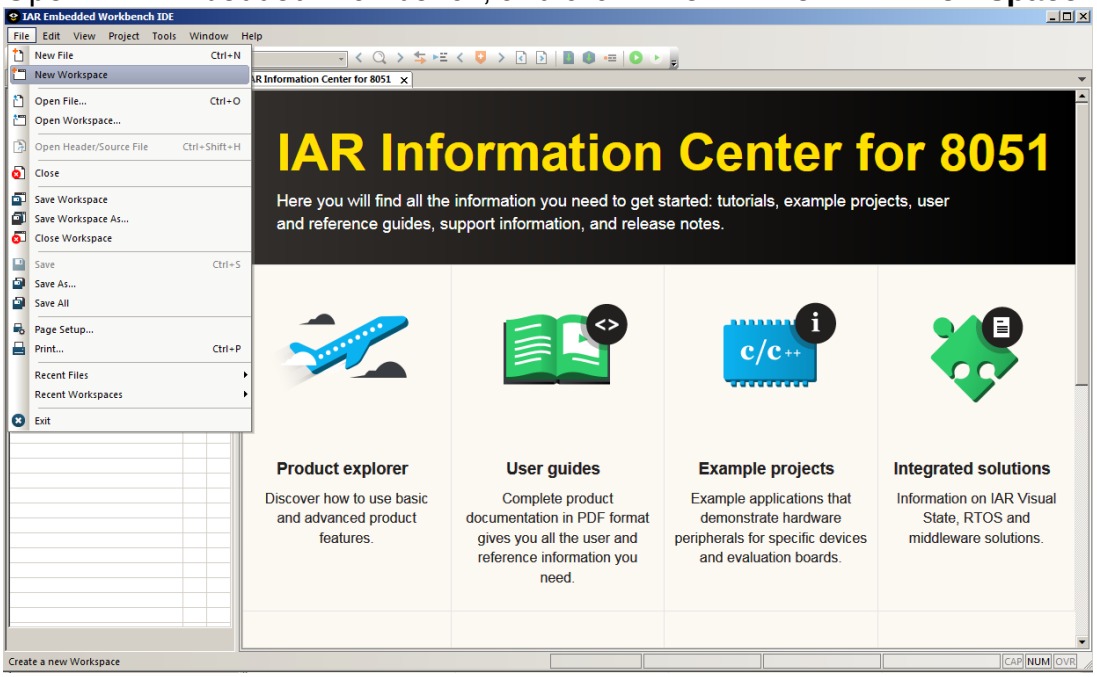

Figure 3.1 Create a New Workspace

2. Create a new project by clicking "**Project**" → "**Create New Project**".

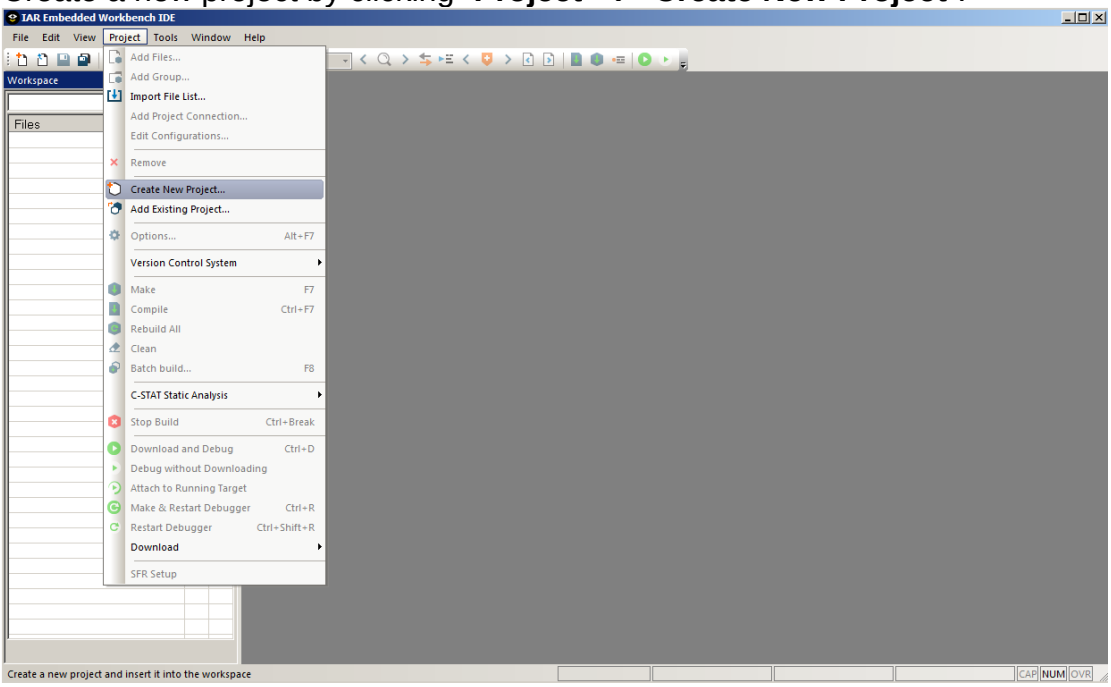

Figure 3.2 Create a New Project

3. Select "ARM" as the tool chain for this project, and then click "**OK**".

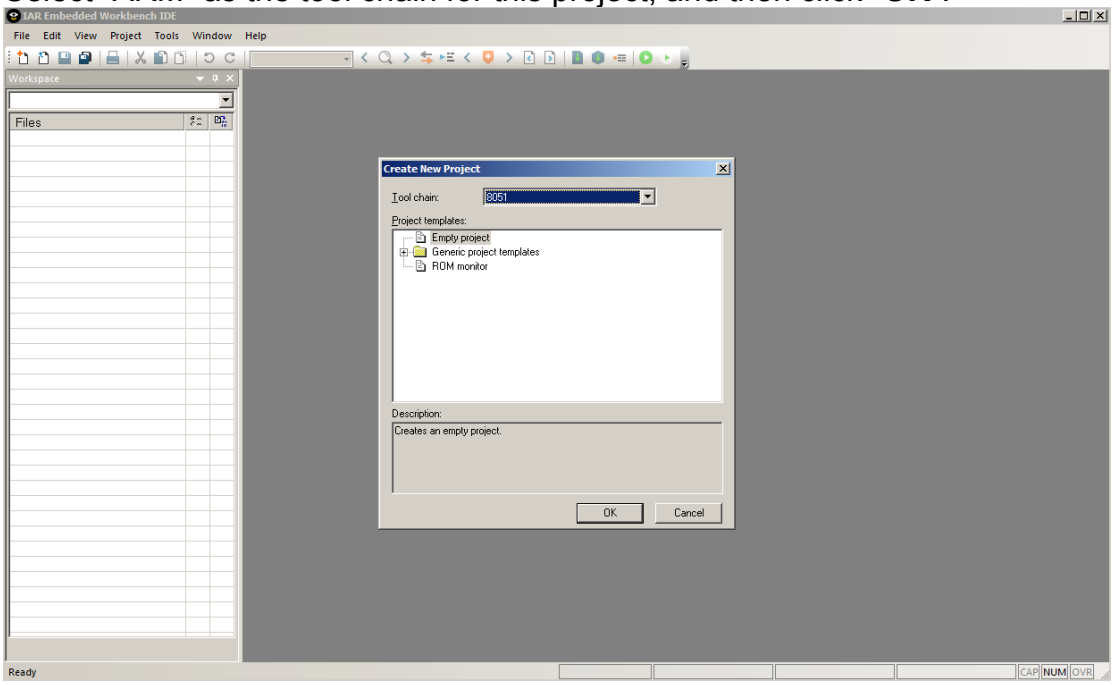

Figure 3.3 Select the Tool Chain for the Project

4. Now you'll be prompted to save the project. Select a folder and input a project name to save it.

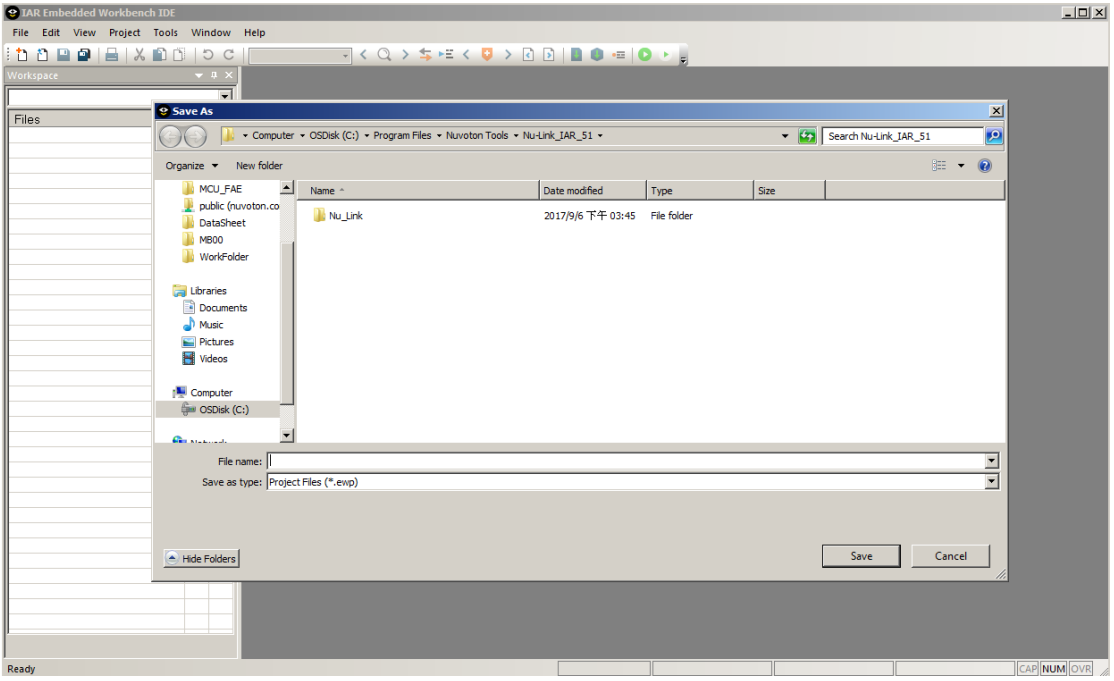

Figure 3.4 Save a Project

5. After the project is saved, right click on the project name of workspace area, and click "**Options**" to open the option setting form.

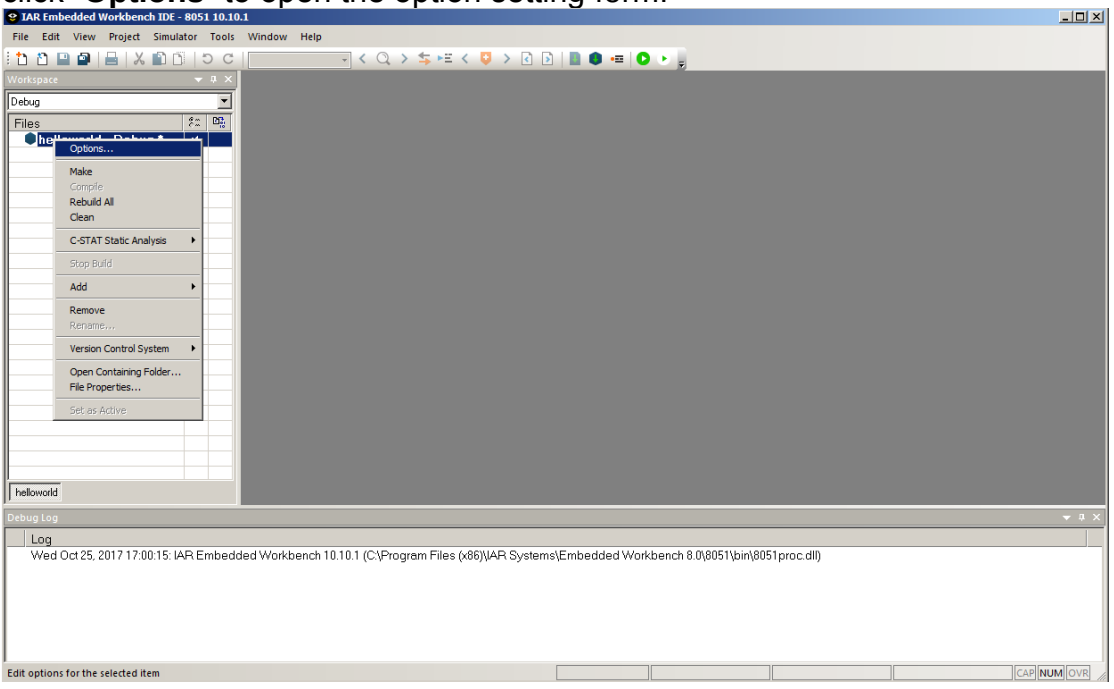

Figure 3.5 Open Project Options Form

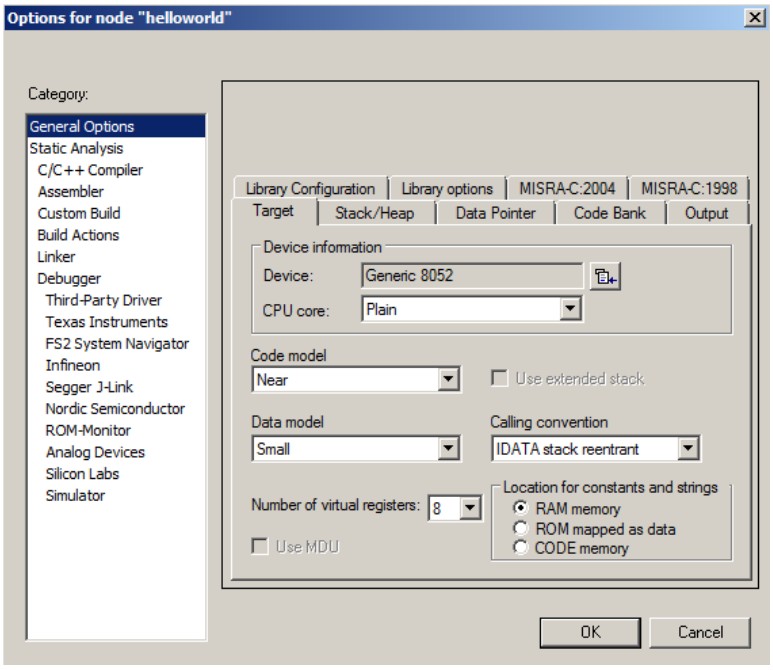

Figure 3.6 Options Form

6. The default options are neither for 8051 series. Check the "**Device**" option and click the "Device" icon on the right to select the correct device name, such as "Nuvoton -> N76E003".

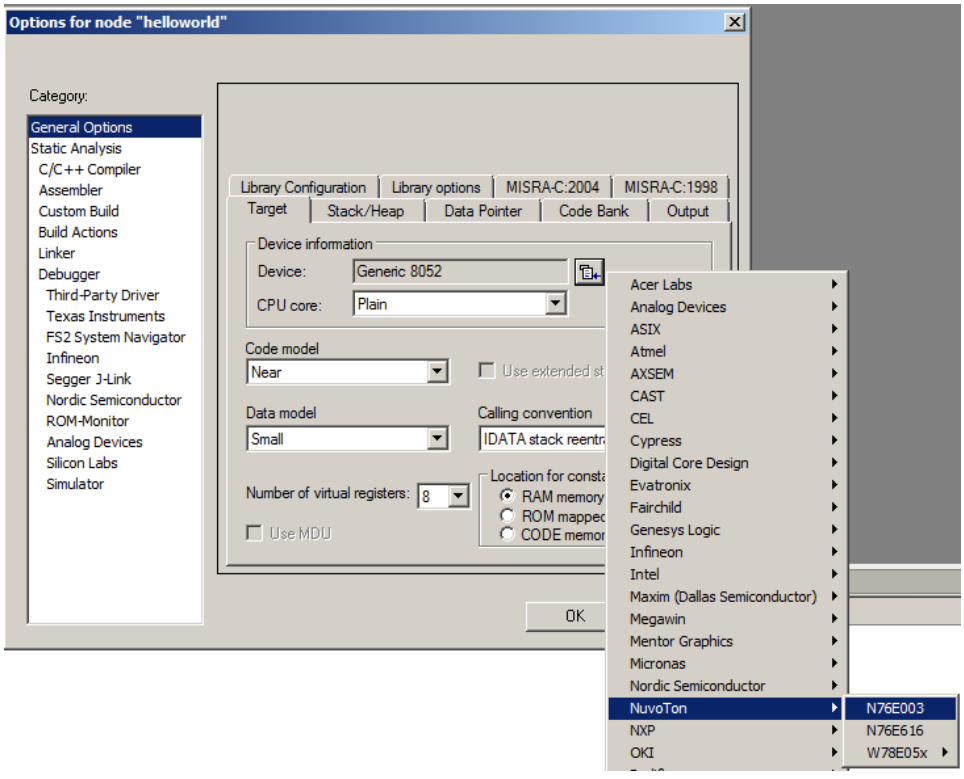

Figure 3.7 Select Target Devices

7. To link and run the program in flash memory, override the default link script for this project. In this example, simply specify the link script used in the sample project "*Samples\GPIO*".

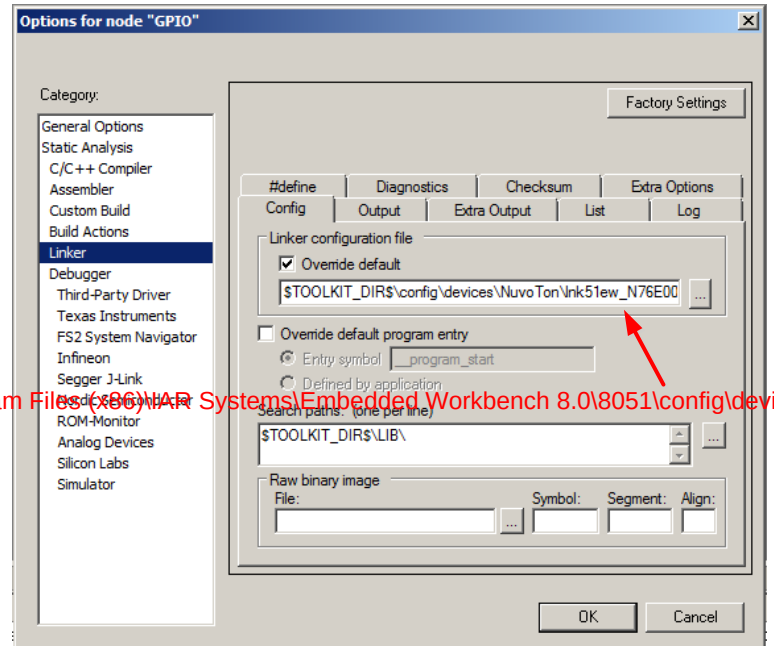

Figure 3.8 Select Linker Configuration File

8. Select "Nu-Link" as the debugger driver for this project. On the "Debugger" page, select "**Third-Party Driver**"; and on the "Third-Party Driver" page, fill in the path of Nu-Link_IAR_51.dll.

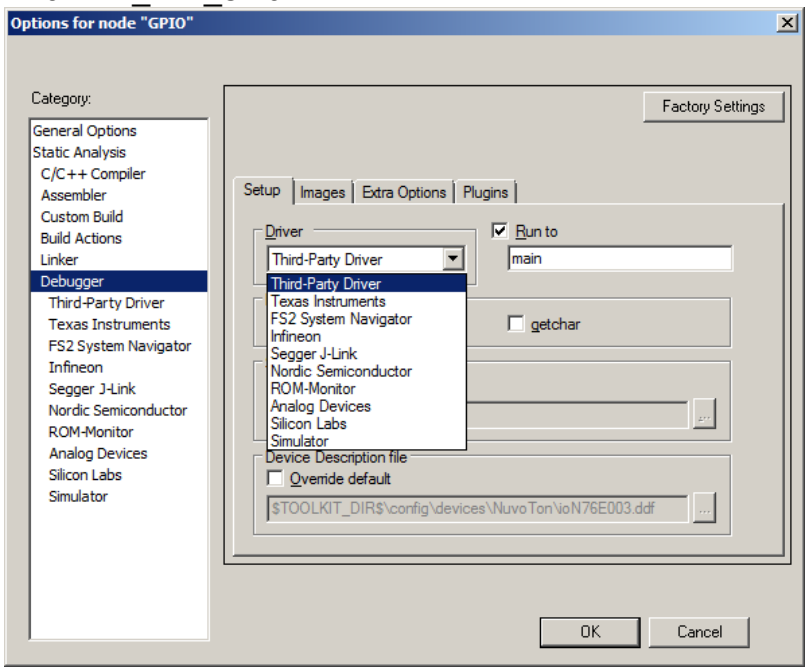

Figure 3.9 Select Debugger Driver

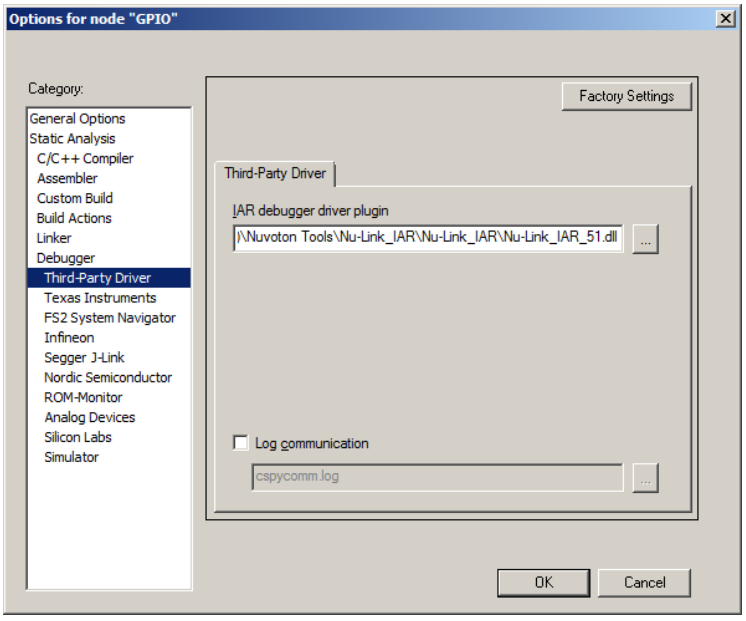

Figure 3.10 Use Nu-Link for Debugger Driver

9. After the option settings are done, click the “OK” button and save the project.

To build the project, you should add the startup code and user application code to the project. Please follow the steps below:

1. Now add the main function to this project. Click “File” → ”New” → “File”, and input the main function in the new file’s editing pane.

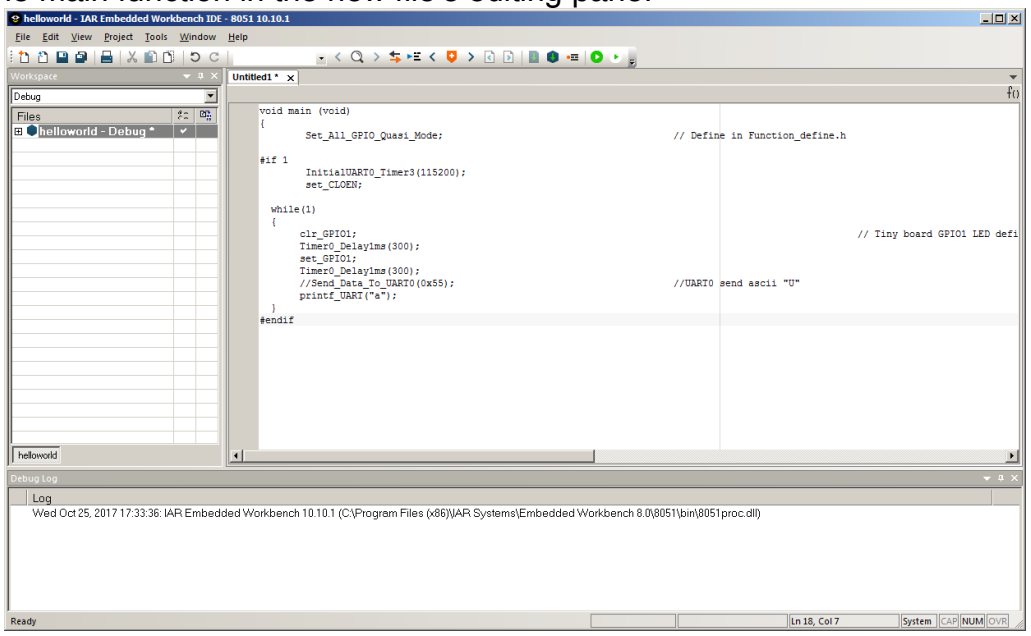

Figure 3.11 Add the Main Function in a New File

2. Click “**File**” → “**Save**” to save the new text file as “*GPIO.c*”.
Add “*GPIO.c*” to the current project. After “*GPIO.c*” has been added, it will be listed in the “Workspace” pane.

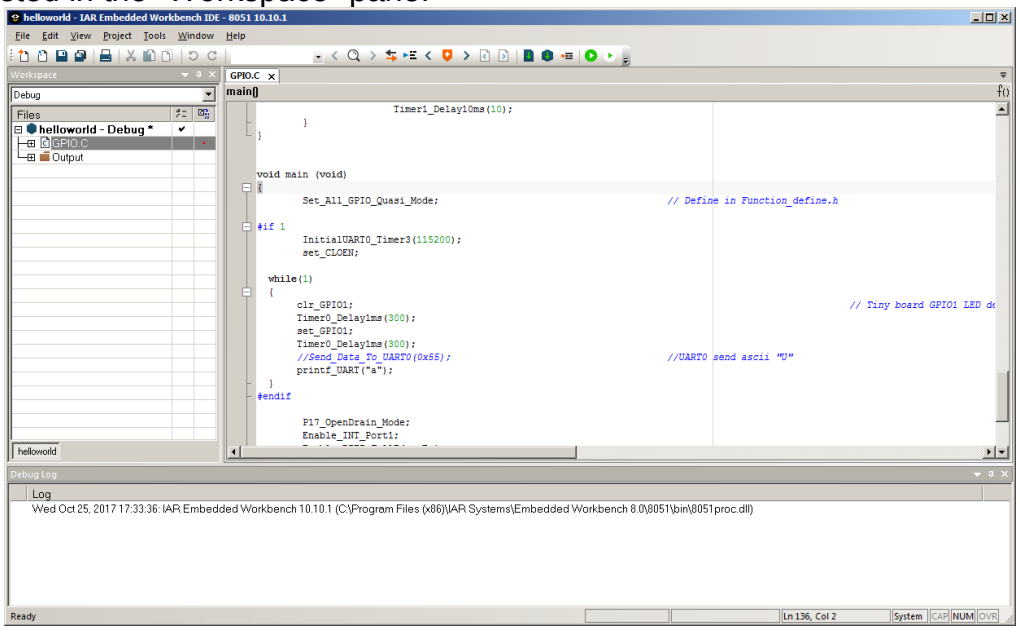

Figure 3.12 Save the New File and add “main.c” File to the Project

3. Click "**File**" → "**Save Workspace**". In the file dialog, input a name for this workspace and save it.

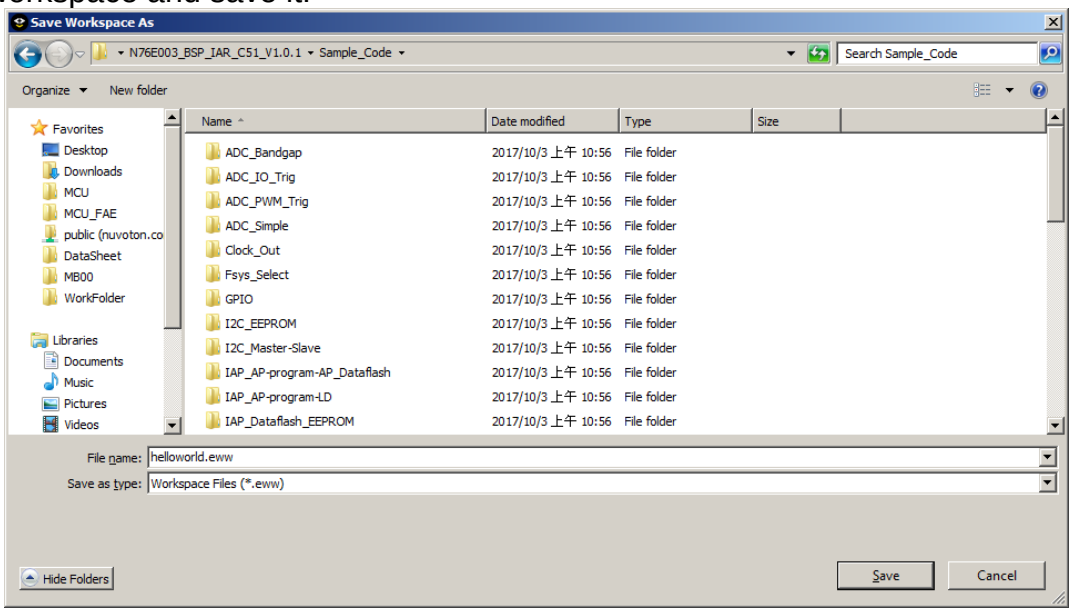

Figure 3.13 Save the Workspace

4. Now start to build the project by clicking "**Project**" → "**Rebuild All**" or "**Make**".

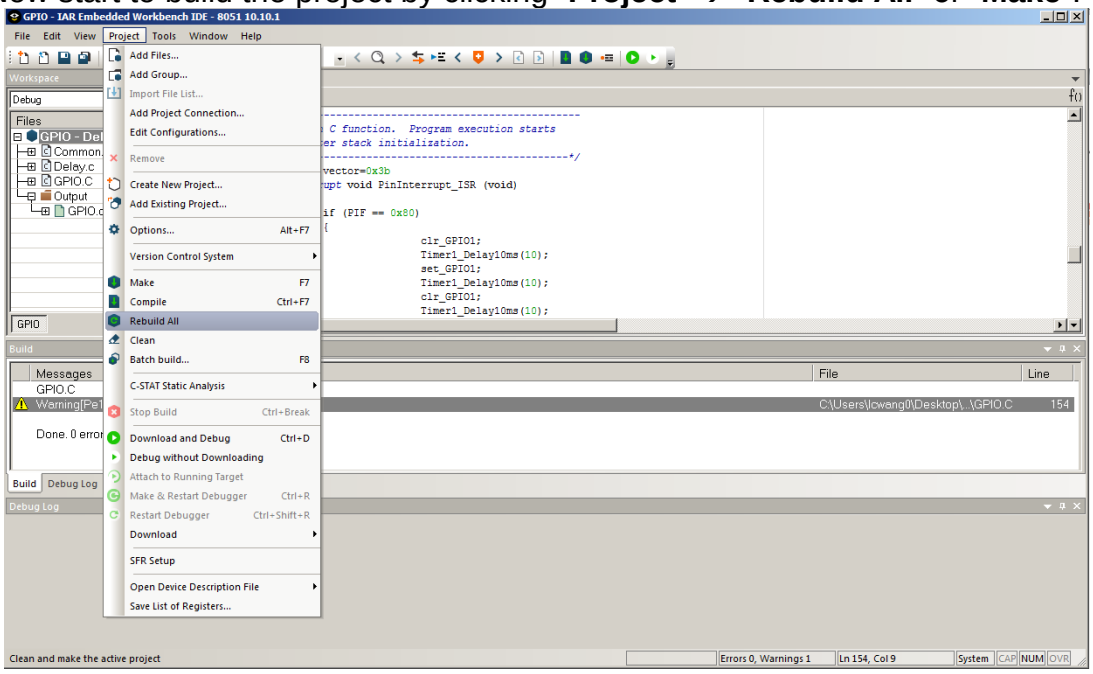

Figure 3.14 Rebuild the Project

5. The project is created successfully.

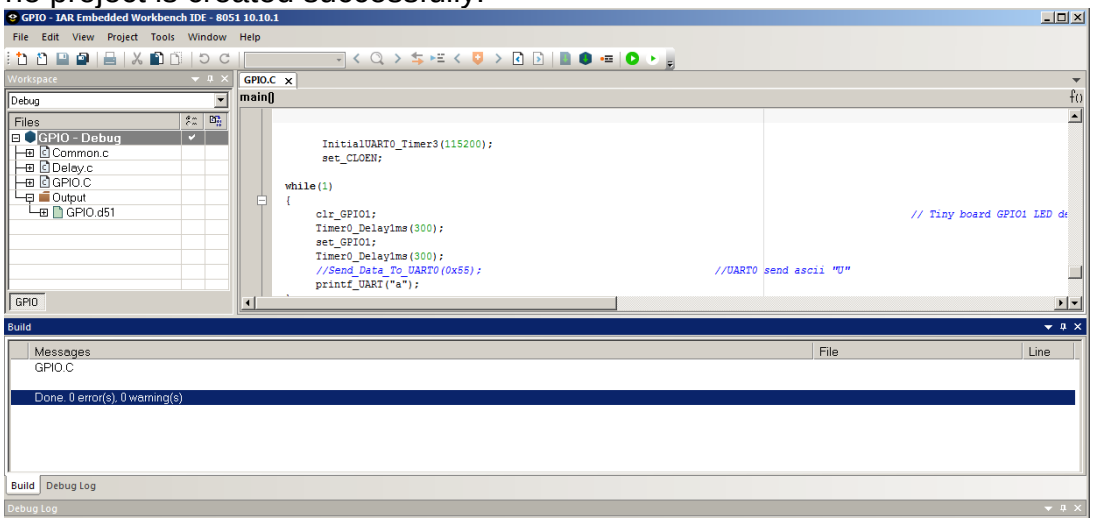

Figure 3.15 Build the Project Successfully

3.2 Debug a project

IAR is a tool for users to debug the project easily. After the project is created successful and the target device and Nu-Link debugger is correctly connected.

- Click "**Project**" → "**Download and Debug**" to start to download and debug the project. Using the project described 3.1, the program will pause at function main().

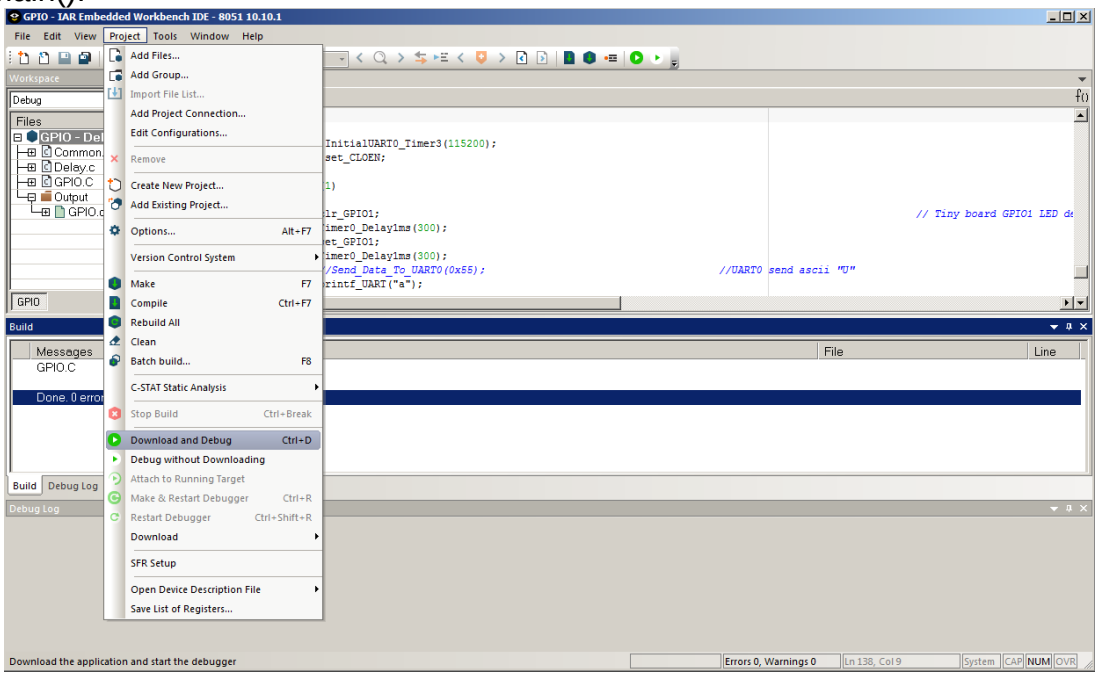

Figure 3.16 Download Code and Debug

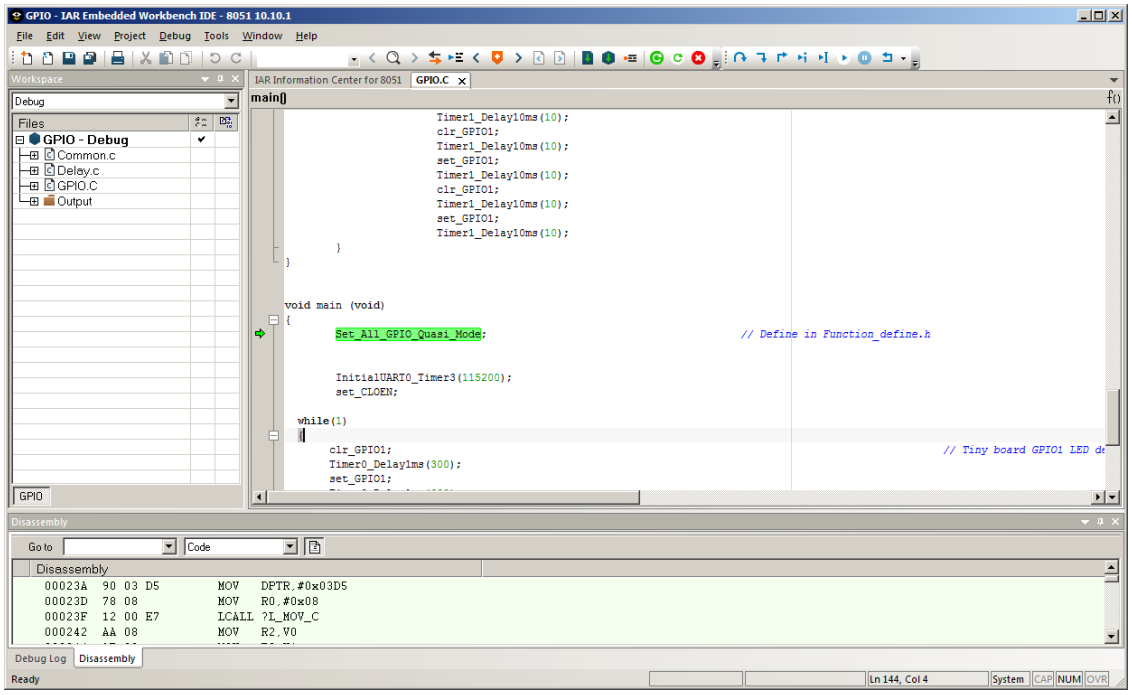

Figure 3.17 Pause at Main Function while Debugging

The following section introduces the debug features using the project described above.

- Move the cursor to the line want to set breakpoint, press "**F9**" to set a breakpoint, and then press "**F5**" to run the program until one breakpoint occurs. Now the program pauses at the breakpoint line.

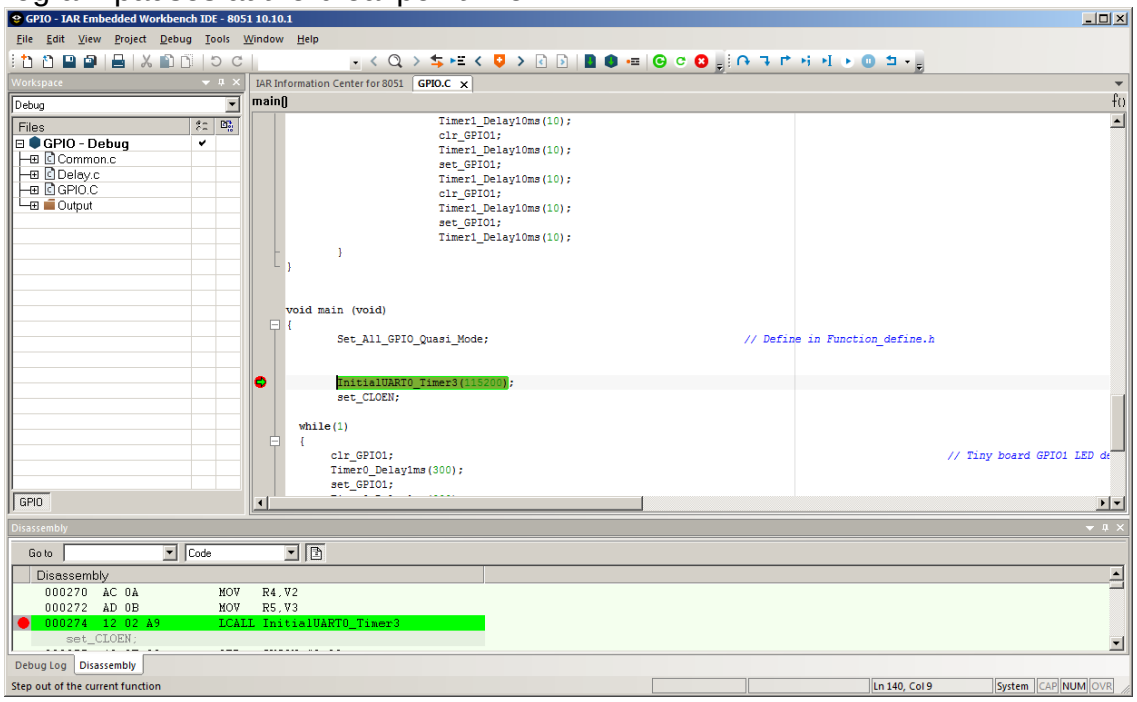

Figure 3.18 Pause the Program by Setting Breakpoints

- Click “**View**” → “**Memory**” to open the Memory pane, in which user can view or edit the memory on the target device.

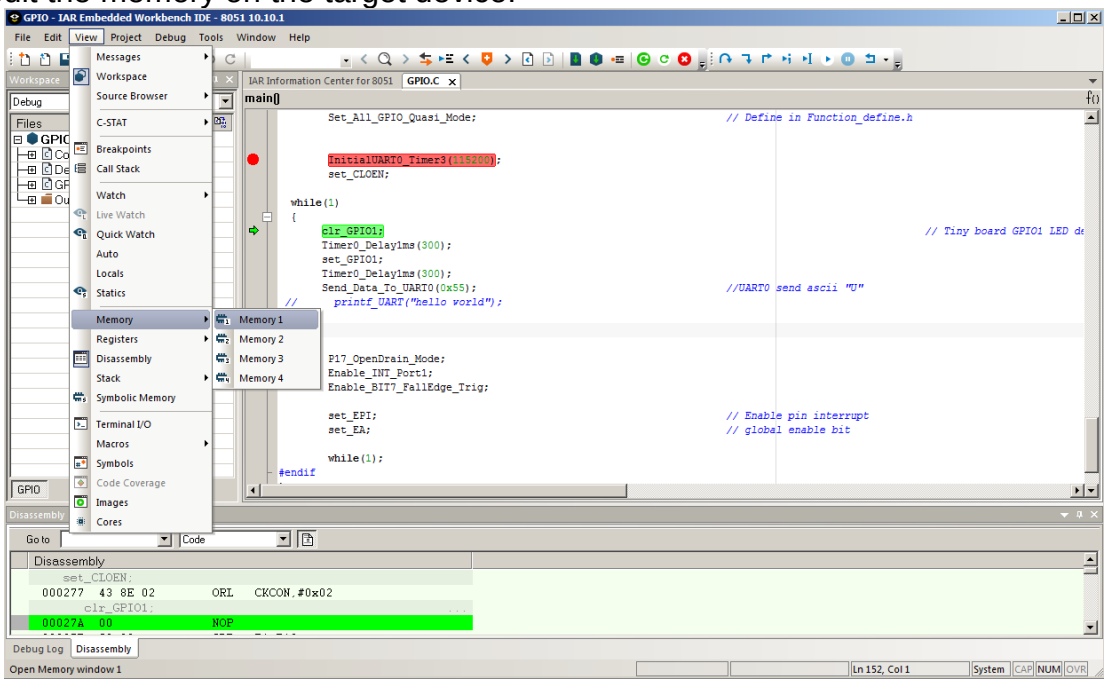

Figure 3.19 Open Memory Pane

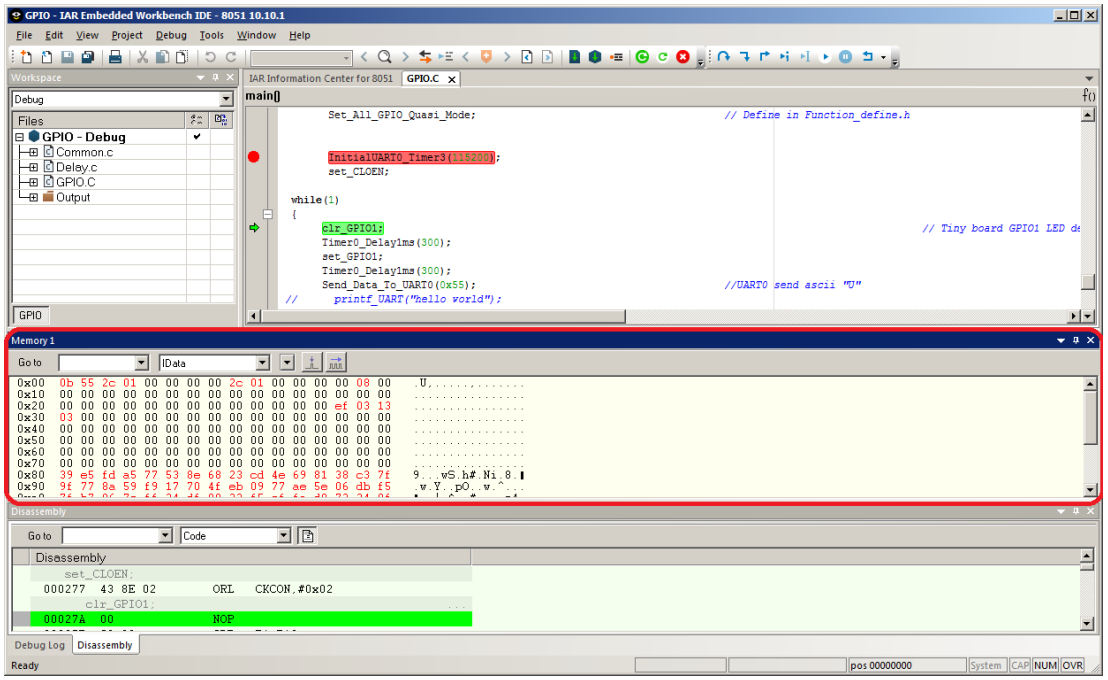

Figure 3.20 Memory Pane

- Click “**View**” → “**Register**” to open the Register pane, in which user can view or edit the registers on the target device.

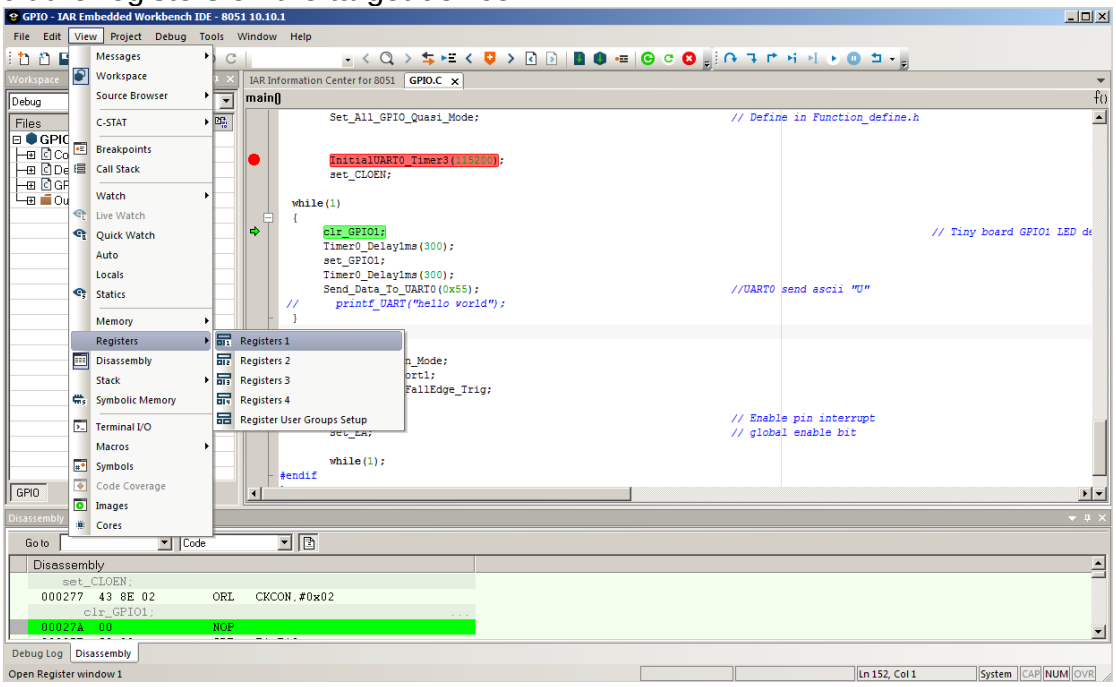

Figure 3.21 Open Register Pane

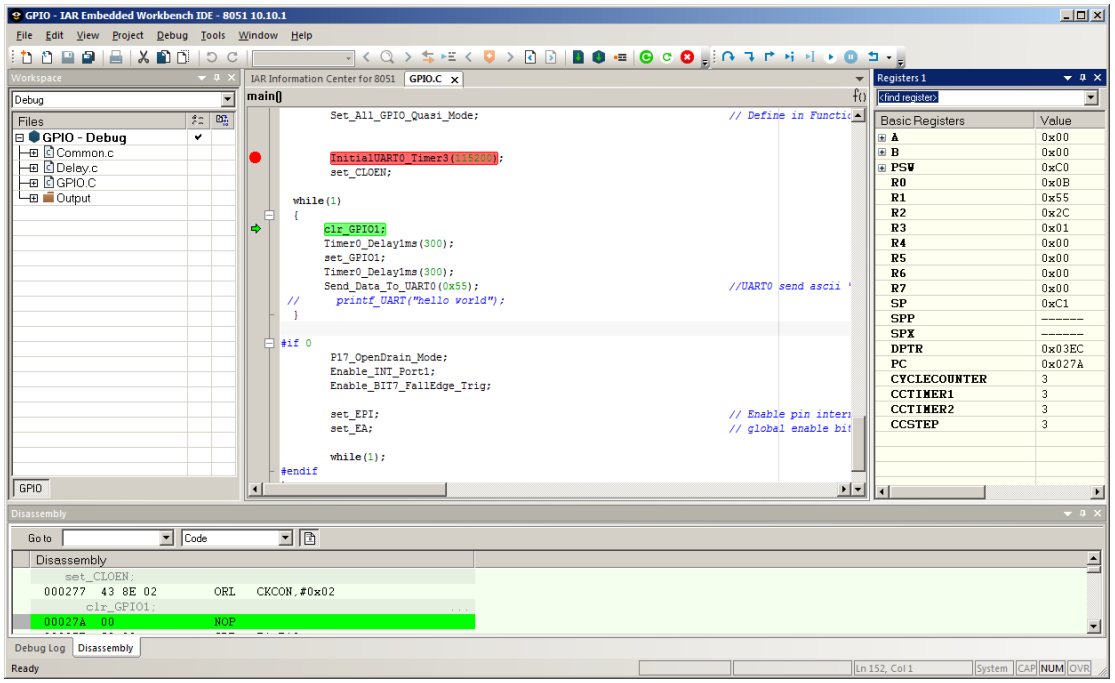

Figure 3.22 Register Pane

4 Firmware Update

When trying to debug a project, it will check the firmware version first. If the current firmware version is not consistent with the installed Nu-Link IAR Driver, a dialog box will pop up as follows:

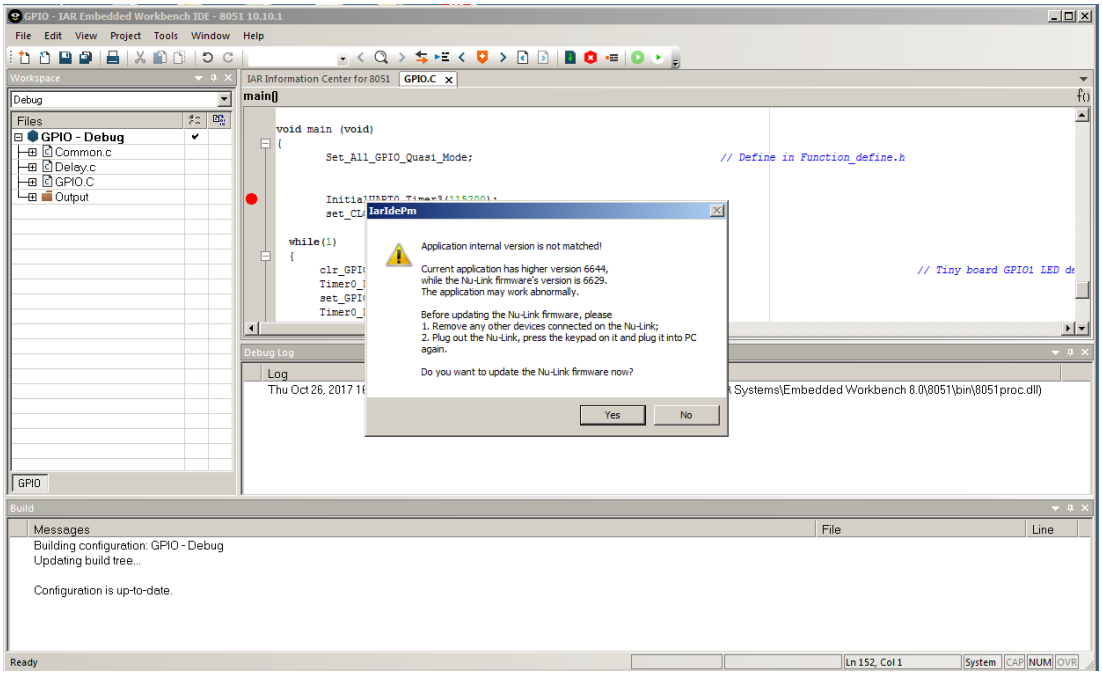

Figure 4.1 Firmware Update Selection Dialog Box

Click "**Yes**" to update firmware or click "**No**" to cancel.

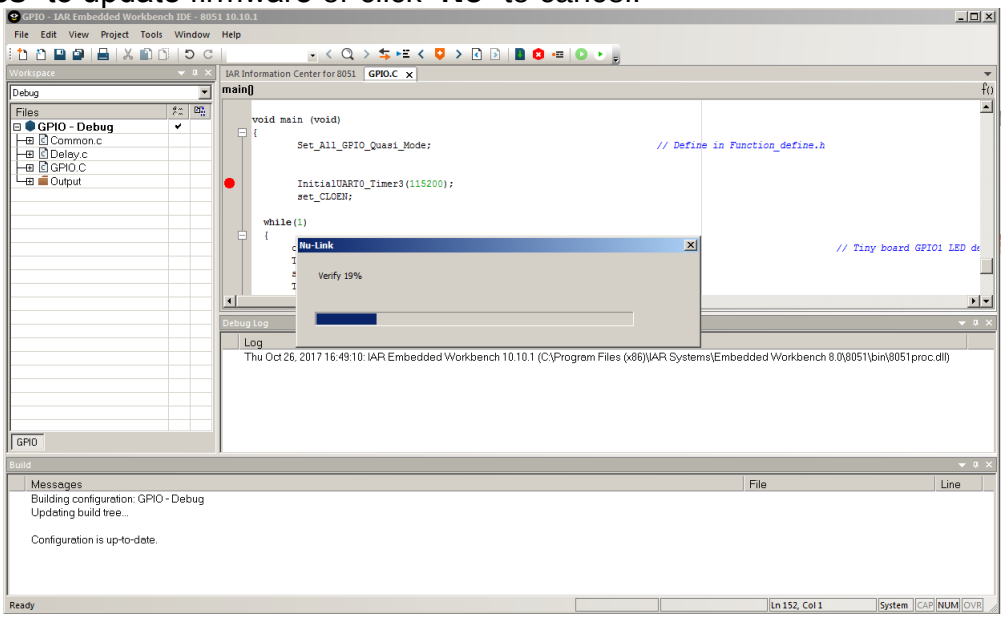

Figure 4.2 Updating Firmware

When update is complete, it is necessary to recreate a connection between Nu-Link and PC. Please plug out the Nu-Link from PC and plug in again.

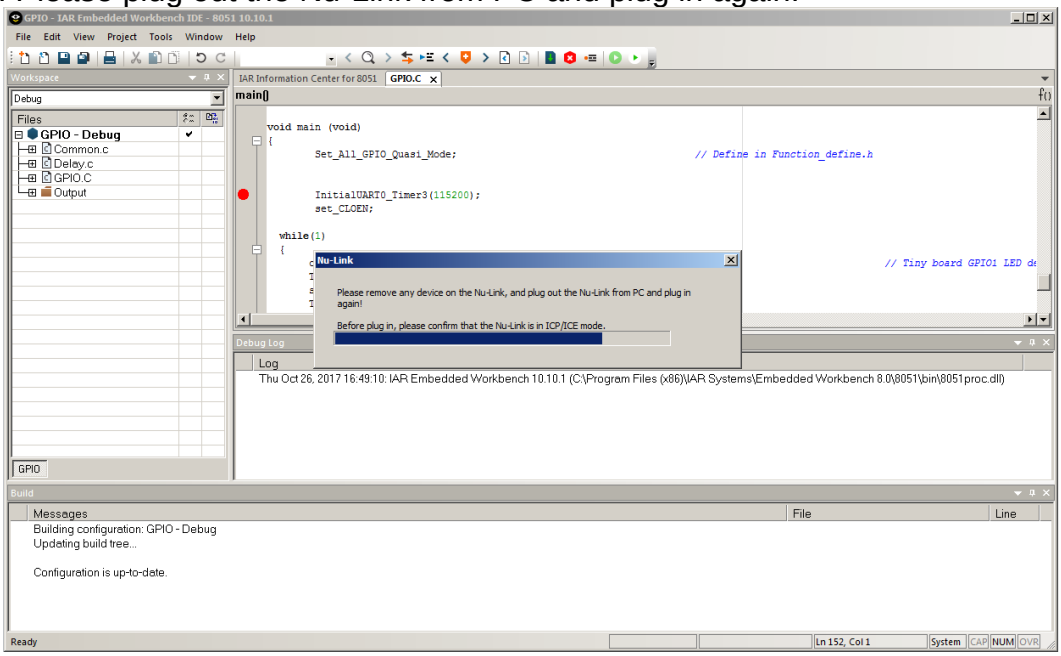

Figure 4.3 Re-connect Nu-Link to Complete Firmware Update

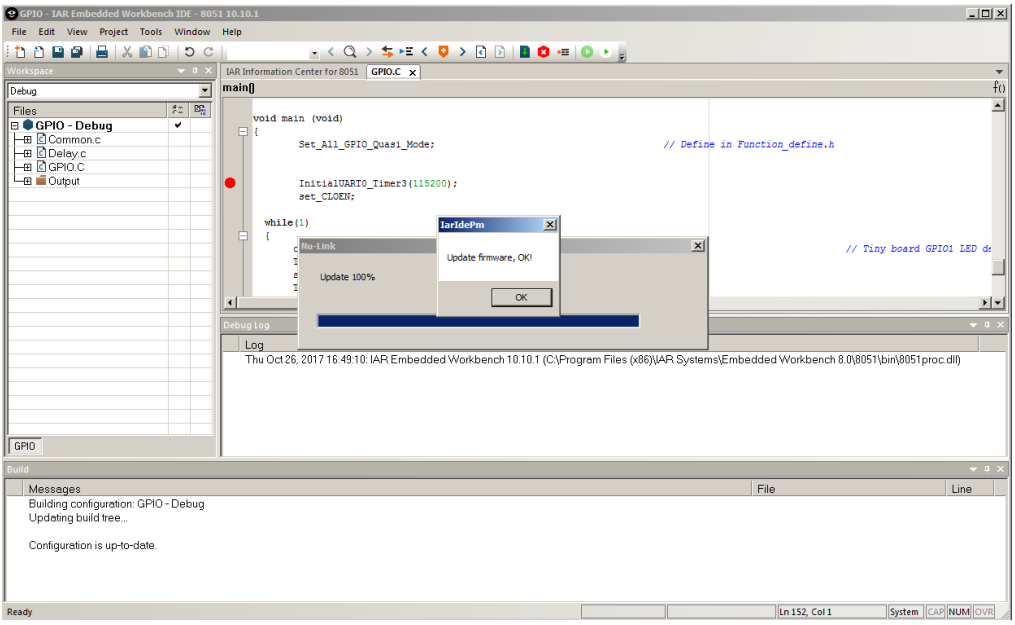

Figure 4.4 Update Firmware Completely

5 Revision History

Revision	Date	Description
1.00	2017/10/25	initial release