

---

## Getting started with STM32L476G discovery kit software development tools

---

### Introduction

This document describes the software environment recommendations required to build an application using the STM32L476G discovery kit (32L476DISCOVERY).

The document provides guidelines to the user on how to build and run a simple example and how to create and build his own application. It has the following structure:

- The first chapter presents the software and hardware requirements (some toolchains supporting the STM32 families, ST-LINK/V2-1 installation and firmware package presentation).
- The second chapter provides step by step guideline on how to execute and debug an application example using some toolchains:
  - IAR Embedded Workbench® for ARM® (EWARM) by IAR Systems®
  - Microcontroller development kit for ARM® (MDK-ARM™) by Keil®
  - TrueSTUDIO® by Atollic®

Although this user manual does not cover all the topics relevant to software development environment, it demonstrates the basic steps necessary to get started with the compilers/debuggers and includes references for complementary information.

# Contents

<b>1</b>	<b>System requirements</b>	<b>5</b>
<b>2</b>	<b>IDEs supporting STM32 families</b>	<b>7</b>
<b>3</b>	<b>ST-LINK/V2-1 installation</b>	<b>8</b>
<b>4</b>	<b>Firmware package</b>	<b>9</b>
<b>5</b>	<b>Executing and debugging firmware using a software toolchains</b>	<b>10</b>
5.1	EWARM toolchain	10
5.2	MDK-ARM toolchain	12
5.3	TrueSTUDIO toolchain	14
<b>6</b>	<b>SW toolchains helpful references and links</b>	<b>20</b>
<b>7</b>	<b>Revision history</b>	<b>21</b>

List of tables

Table 1. Toolchains supporting STM32L476G Discovery ..... 7

Table 2. Links to software toolchains ..... 20

Table 3. Document revision history ..... 21

## List of figures

Figure 1.	Hardware environment . . . . .	5
Figure 2.	Package contents . . . . .	9
Figure 3.	IAR Embedded Workbench IDE . . . . .	10
Figure 4.	EWARM project successfully compiled. . . . .	10
Figure 5.	Download and Debug button . . . . .	11
Figure 6.	IAR Embedded Workbench debugger screen . . . . .	11
Figure 7.	EWARM Go button . . . . .	12
Figure 8.	uVision5 IDE . . . . .	12
Figure 9.	MDK-ARM project successfully compiled . . . . .	13
Figure 10.	Start/Stop Debug Session button . . . . .	13
Figure 11.	MDK-ARM debugger screen. . . . .	14
Figure 12.	Run button . . . . .	14
Figure 13.	TrueSTUDIO® workspace launcher dialog box . . . . .	15
Figure 14.	TrueSTUDIO import source select dialog box . . . . .	16
Figure 15.	TrueSTUDIO import projects dialog box. . . . .	17
Figure 16.	TrueSTUDIO® project successfully compiled . . . . .	18
Figure 17.	TrueSTUDIO debug window. . . . .	19

# 1 System requirements

Before running the application, execute the following steps:

- Install the preferred Integrated Development Environment (IDE)
- Install the ST-LINK V2-1 driver from the ST Website
- Download the STM32L476G Discovery firmware from the ST Website
- Establish the connection with the STM32L476G Discovery board as showed in [Figure 1: Hardware environment](#)

**Figure 1. Hardware environment**



**The above steps will be detailed in the following sections.**

To run and develop any firmware application on STM32L476G Discovery board, the minimum requirements are as follows:

- Windows PC (XP, Vista, 7,8)
- “USB type A to Mini-B” cable, used to power the board (through USB connector CN1) from host PC and connect to the embedded ST-LINK/V2-1 for debugging and programming

## 2 IDEs supporting STM32 families

STM32 families of 32-bit ARM® Cortex® -M core-based microcontrollers are supported by a complete range of software tools, such as traditional integrated development environments IDEs with C/C++ compilers and debuggers from major 3rd-parties (free versions up to 64 KB of code, depending on partner), completed with innovative tools from STMicroelectronics.

The following table regroups general information about the most used integrated development environments, as well as the version officially supporting STM32L476 product.

**Table 1. Toolchains supporting STM32L476G Discovery**

Toolchain	Company	Compiler	Version	Download link <sup>(1)</sup>
EWARM	IAR Systems®	IAR C/C++	7.40 and later	<a href="http://www.iar.com">www.iar.com</a> 30-day evaluation edition KickStart edition(16KB Limitation for Cortex M0)
MDK-ARM	Keil™	ARMCC	5.xx and later with Keil.STM32L4xx_ DFP1.0.0.pack	<a href="http://www.keil.com">www.keil.com</a> MDK-Lite (32KB Code size limitation)
TrueSTUDIO	Atollic®	GNUC	5.3.1 and later	<a href="http://www.atollic.com">www.atollic.com</a> <sup>(1)</sup> 32KB Limitation (8KB on Cortex-M0 and Cortex-M1) 30 day Professional version (Trial)

1. Registration before download is required.

### 3 ST-LINK/V2-1 installation

The STM32L476G Discovery board includes an embedded ST-LINK/V2 debug tool interface. The interface needs an ST-Link/V2 dedicated USB driver to be installed. This driver is available at ST Website ([www.st.com](http://www.st.com)) and it is supported by the software toolchains:

- IAR Embedded Workbench for ARM (EWARM).
- The toolchain is installed by default in the C:\Program Files\IAR Systems\Embedded Workbench x.x directory on the local hard disk of the PC. After installing EWARM, install the ST-LINK/V2 driver by running the ST-Link\_V2\_USB.exe from [IAR\_INSTALL\_DIRECTORY]\Embedded Workbench x.x\arm\drivers\ST-Link \ST-Link\_V2\_USBdriver.exe.
- Keil Microcontroller Development Kit (MDK-ARM) toolchain. The toolchain is installed by default in the C:\Keil directory on the local hard disk of the PC; the installer creates a µVision4 shortcut in the start menu. When connecting the ST-LINK/V2 tool, the PC detects the new hardware and prompts the user to install the ST-LINK\_V2\_USB driver. The “Found New Hardware Wizard” displays and guides the user through the steps required to install the driver from the recommended location.
- Atollic TrueSTUDIO STM32.
- The toolchain is installed by default in the C:\Program Files\Atollic directory on the local hard disk of the PC. The ST-Link\_V2\_USB.exe is installed automatically when installing the software toolchain.

Complementary information on the firmware package content and the STM32L476 Discovery requirements is available inside the User Manual UM1879: Discovery kit with STM32L476VG MCU.

*Note:* The embedded ST-LINK/V2-1 supports only the SWD interface for STM32 devices.



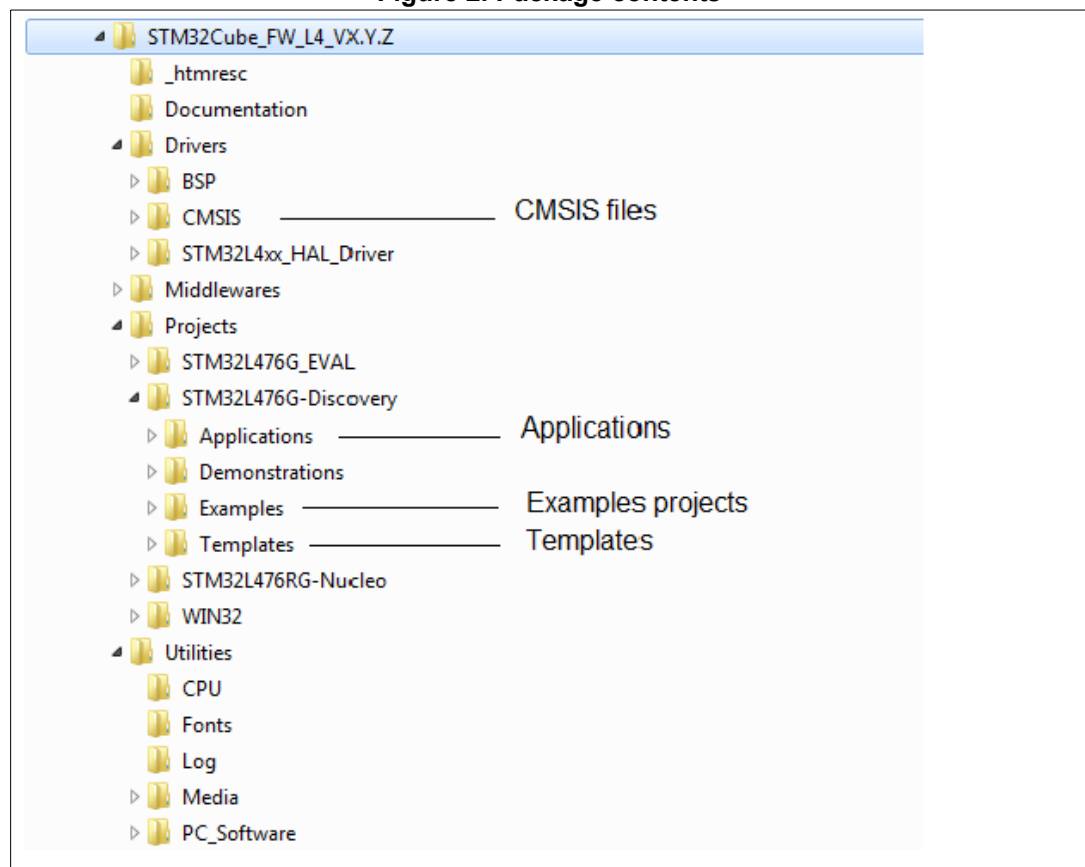
## 4 Firmware package

The STM32L476G Discovery firmware applications, demonstration and IP examples are provided in one single package and supplied in one single zip file. The extraction of the zip file generates a folder named “STM32Cube\_FW\_L4\_VX.Y.Z”, which contains the following subfolders:

- **Template project:** pre-configured project with empty main function to be customized. This is helpful to start creating the user’s application based on the peripherals drivers.
- **Examples:** Including set of examples for each peripheral ready to be run.

See the below [Figure 2](#):

**Figure 2. Package contents**



## 5 Executing and debugging firmware using a software toolchains

### 5.1 EWARm toolchain

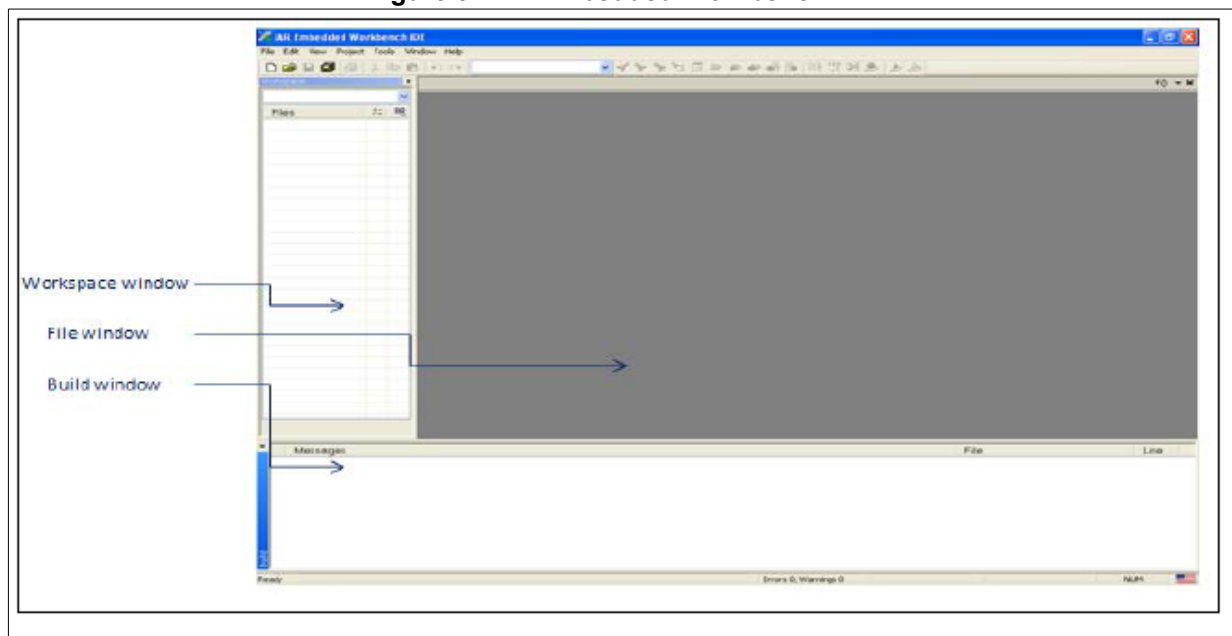
This is the procedure for compiling/linking and executing an existing EWARm project.

The below steps can be applied to an already existing example, demonstration or template project available on the STM32L476G Discovery package firmware web page at [www.st.com](http://www.st.com) web site.

First of all, the user needs to go through firmware/readme.txt file which contains a description of the firmware and hardware/software requirements:

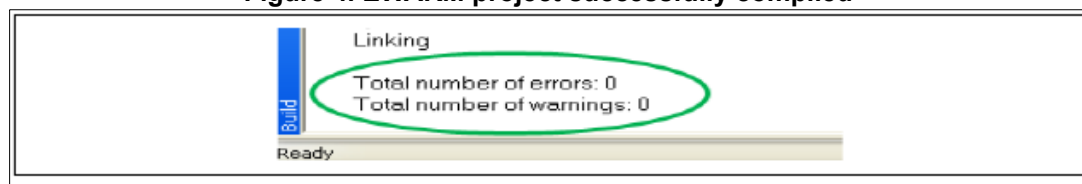
1. Open the IAR Embedded Workbench for ARM (EWARM). *Figure 3* shows the basic names of the windows referred to in this document.

**Figure 3. IAR Embedded Workbench IDE**



2. In the **File** menu, select **Open** and click **Workspace** to display the Open Workspace dialog box. Browse to select either an example or demonstration or template workspace file, and click Open to launch it in the Project window.
3. In the **Project** menu, select **Rebuild All** to compile the project.
4. If the **Project** is successfully compiled, the window shown in *Figure 4: EWARm project successfully compiled* is displayed.

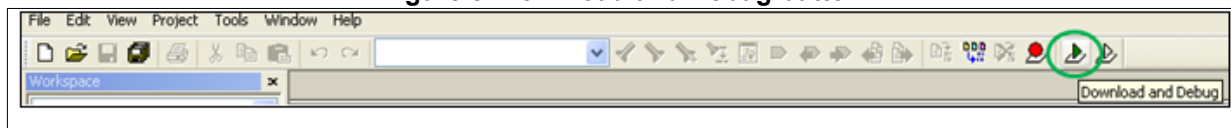
**Figure 4. EWARm project successfully compiled**



If the user needs to change the project settings (Include and preprocessor defines), he must follows the project options:

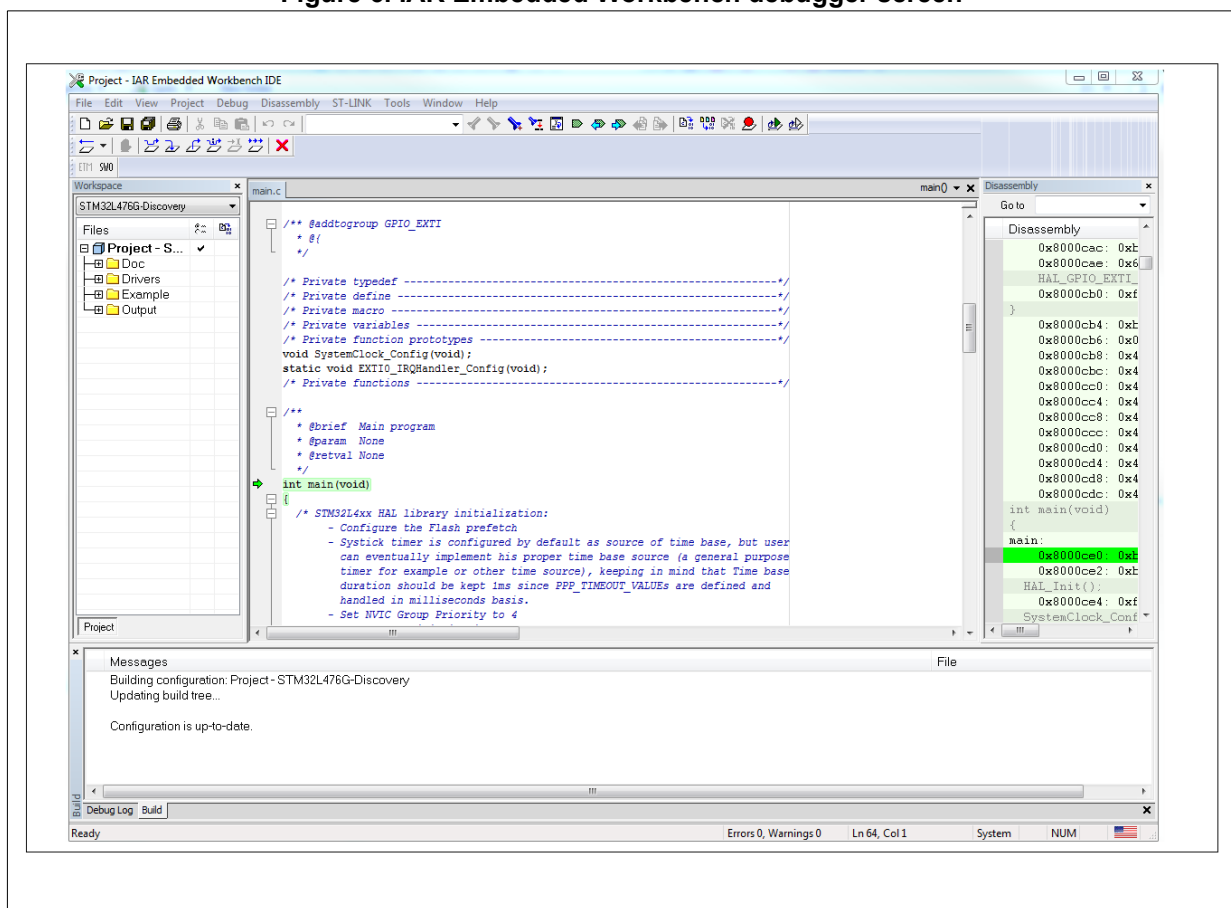
- For Include directories.
  - Project>Options...>C/C++ compiler>.
  - For pre-processor defines.
  - Project>Options...>C/C++ compiler>pre-processor>.
5. In the IAR Embedded Workbench IDE, from the Project menu, select Download and Debug or, alternatively, click the Download and Debug button in the toolbar, to program the Flash memory and begin debugging.

**Figure 5. Download and Debug button**



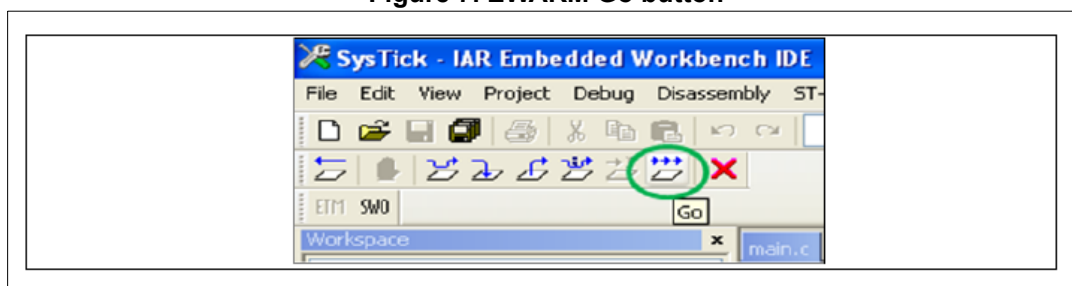
6. The debugger in the IAR Embedded Workbench can be used to debug source code at C and assembly levels, to set breakpoints, to monitor individual variables and watch events during the code execution.

**Figure 6. IAR Embedded Workbench debugger screen**



To run the application, from the Debug menu, select Go. Alternatively, click the Go button in the toolbar to run the application.

Figure 7. EWARM Go button



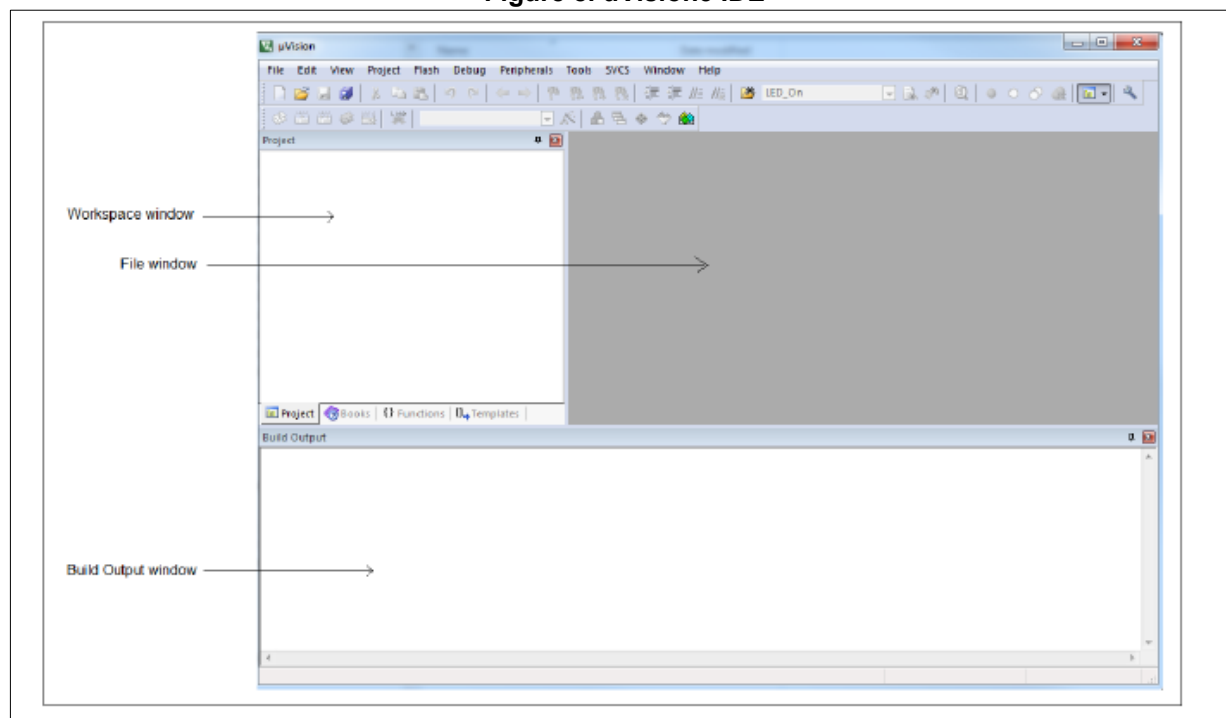
## 5.2 MDK-ARM toolchain

This is the procedure for compiling/linking and executing an existing MDK-ARM project.

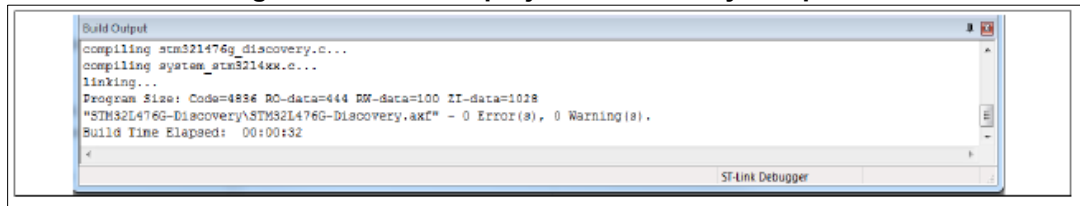
1. Open the Keil MDK-ARM Microcontroller Kit.

*Figure 8* shows the “Keil uVision5” windows.

Figure 8. uVision5 IDE



2. In the **Project** menu, select **Open Project**. Browse to select either an example or demonstration or template project file, and click **Open** to launch it in the Project window.
3. In the **Project** menu, select **Rebuild All target files** to compile the project.
4. If the project is successfully compiled, the following window in *Figure 9: MDK-ARM project successfully compiled* is displayed.

**Figure 9. MDK-ARM project successfully compiled**

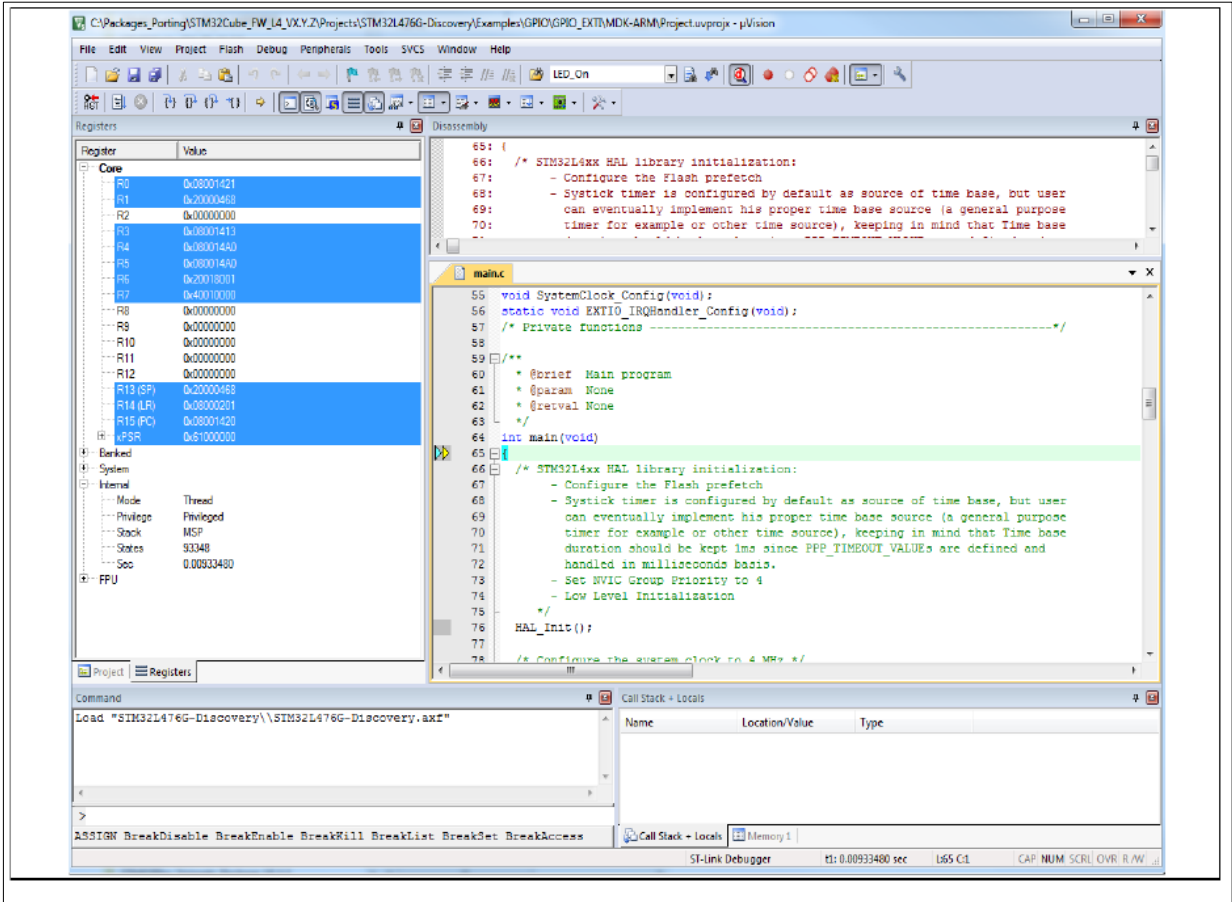
If the user needs to change his project settings (Include and preprocessor defines), he must follow the project options:

- For Include directories.
  - Project>Options for Target > C/C++ > Include Paths.
  - For pre-processor defines.
  - Project>Options for Target > C/C++ > Preprocessor symbols > Define.
5. In the MDK-ARM IDE, from the Debug menu, select Start/Stop Debug Session or, alternatively, click the Start/Stop Debug Session button in the toolbar, to program the Flash memory and begin debugging.

**Figure 10. Start/Stop Debug Session button**

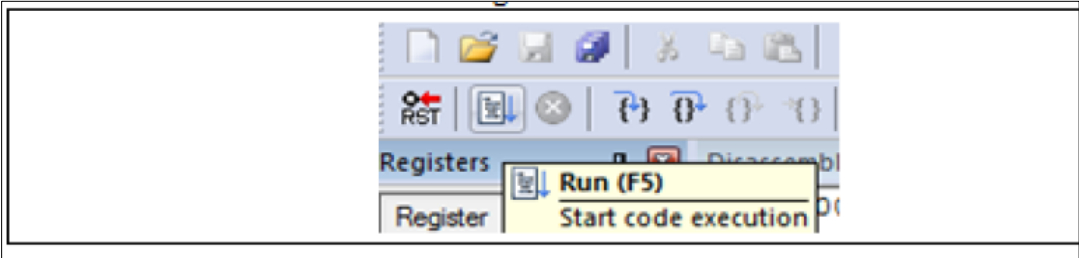
6. The MDK-ARM debugger can be used to debug source code at C and assembly levels, to set breakpoints, to monitor individual variables and to watch events during the code execution.

Figure 11. MDK-ARM debugger screen



To run the application, from the Debug menu, select Run. Alternatively, click the Run button in the toolbar to run the application.

Figure 12. Run button



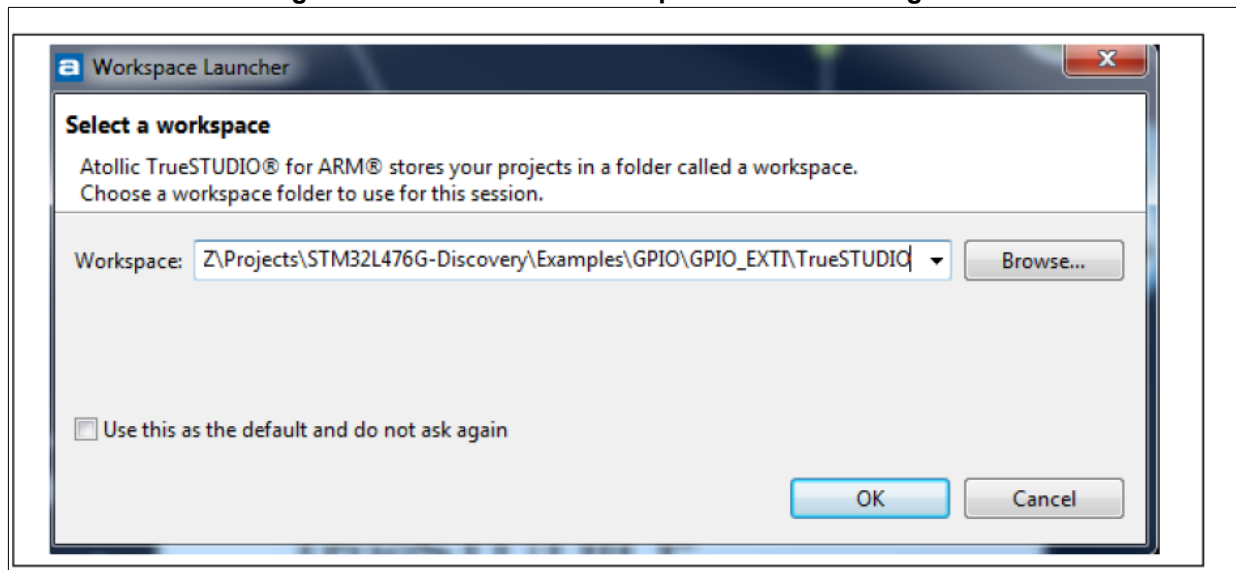
5.3 TrueSTUDIO toolchain

This is the procedure for compiling/linking and executing an existing TrueSTUDIO project.



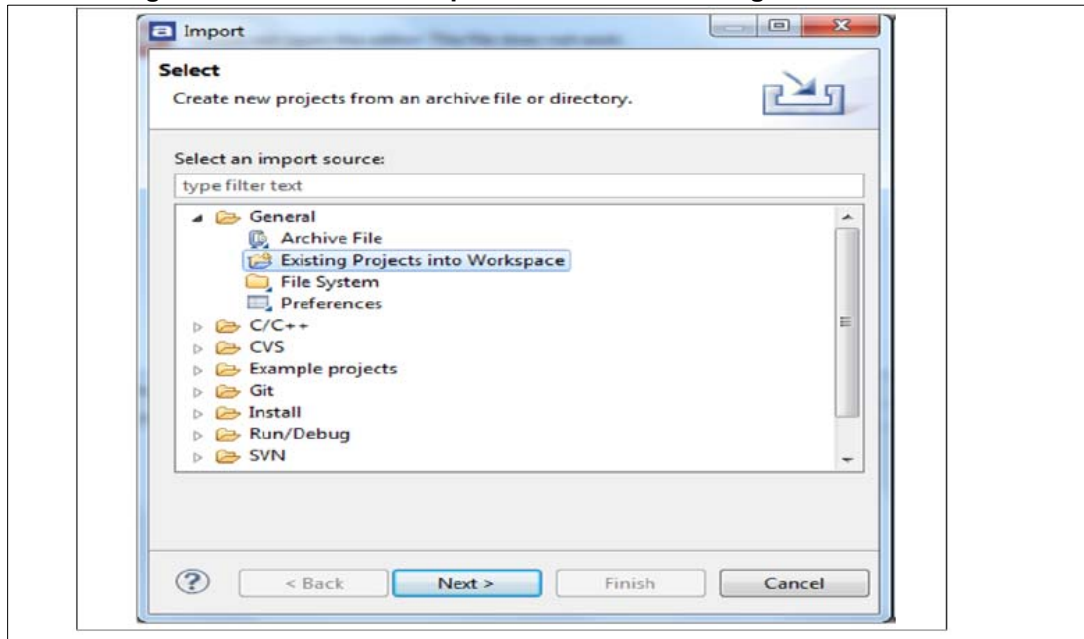
1. Open the Atollic TrueSTUDIO for ARM product. The program launches and prompts for the Workspace location as showed in [Figure 13](#).

**Figure 13. TrueSTUDIO® workspace launcher dialog box**



2. Browse to select a TrueSTUDIO workspace of either an example or demonstration or template workspace file and click OK to load it.
3. To load an existing project in the selected workspace, select Import from the File menu to display the Import dialog box.
4. In the Import window, open General, select Existing Projects into Workspace and click Next (see [Figure 14](#)).

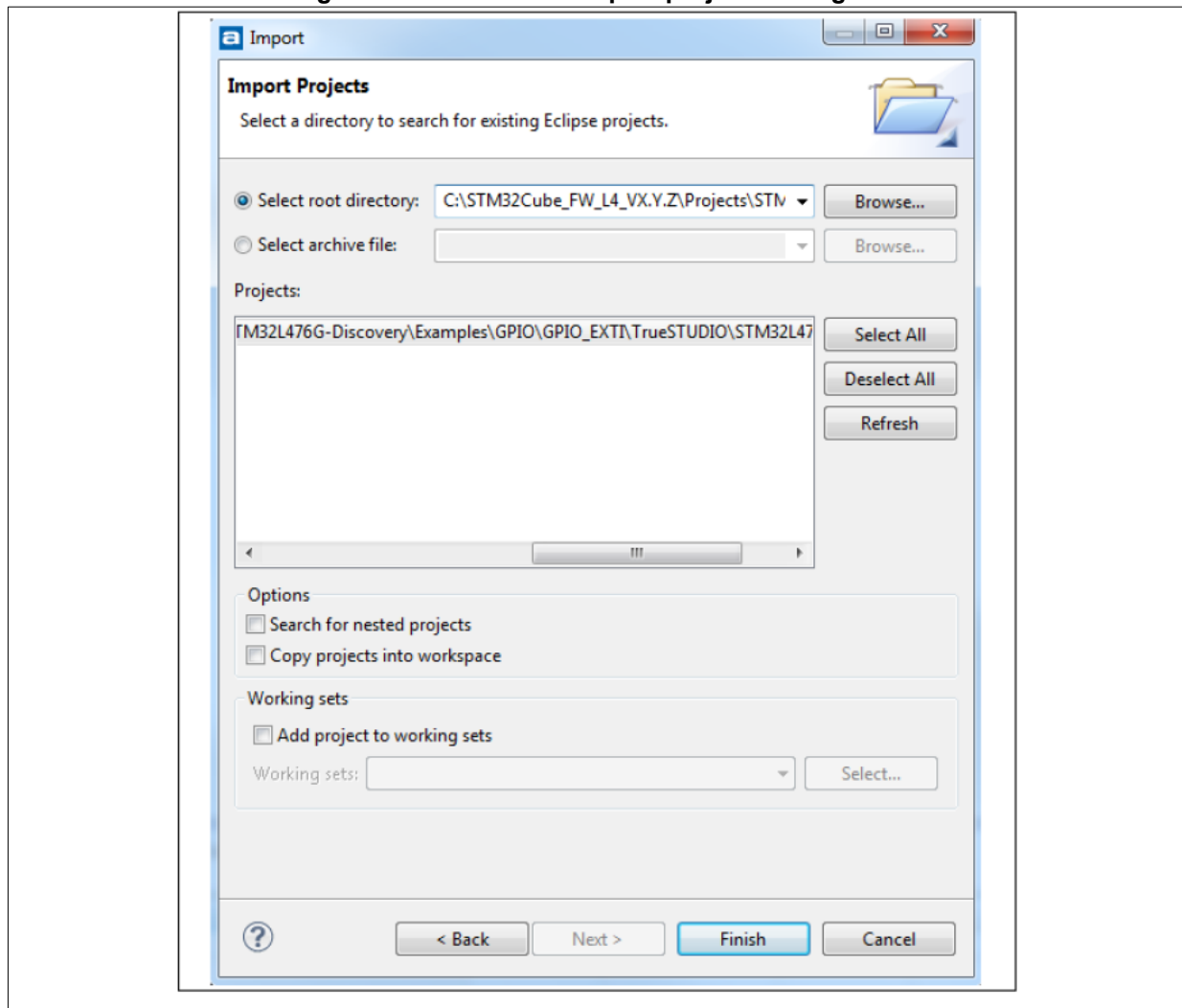
Figure 14. TrueSTUDIO import source select dialog box



5. Click Select root directory, browse to the TrueSTUDIO workspace folder (see [Figure 15](#)).

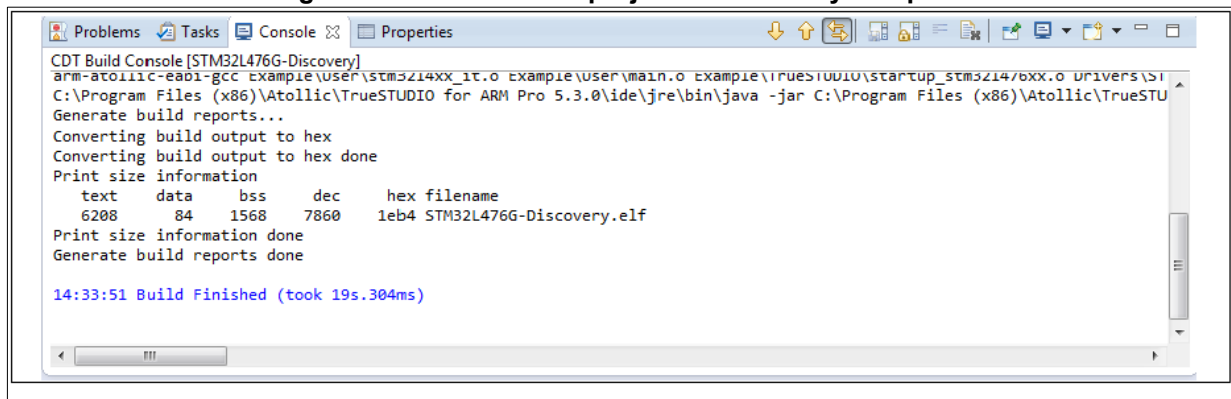


Figure 15. TrueSTUDIO import projects dialog box



6. In the Projects panel, select the project and click Finish.
7. In the Project Explorer, select the project, open the Project menu, and click Build Project.
8. If the project is successfully compiled, the following messages display on the Console window (see [Figure 16](#)).

Figure 16. TrueSTUDIO® project successfully compiled

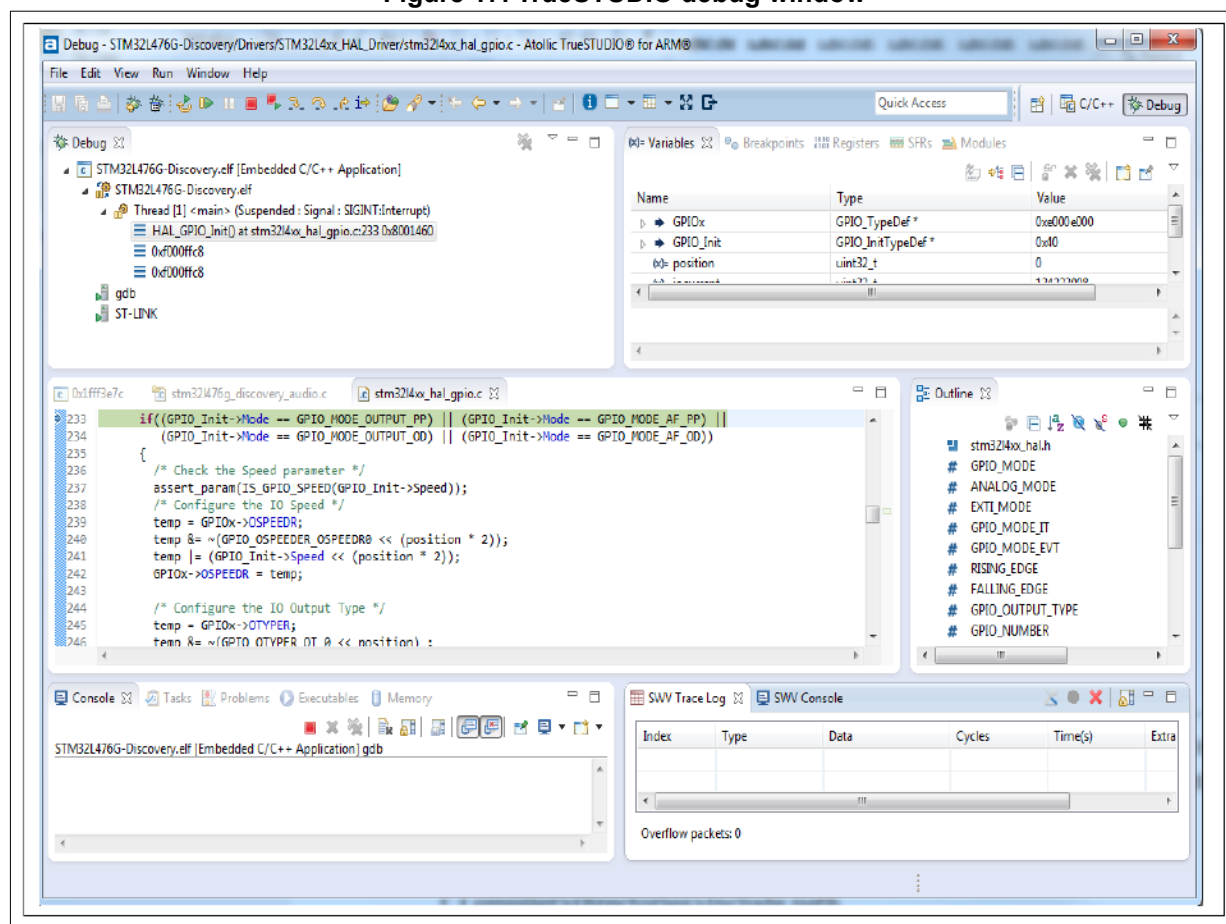


If the user needs to change the project settings (Include directories and preprocessor defines), he must follow Project>Properties, select C/C++ Build>Settings from the left panel:

- For Include directories  
C Compiler>Directories>Include path
  - For pre-processor defines  
C Compiler>Symbols> Defined symbols
9. To debug and run the application, select the project In the Project Explorer and **press F11** to start a debug session.

In the Project Explorer, select the project and press F11 to start a debug session (see [Figure 17: TrueSTUDIO debug window](#)).

Figure 17. TrueSTUDIO debug window



The debugger in the Atollic TrueSTUDIO can be used to debug source code at the C and assembly levels, to set breakpoints, to monitor individual variables and to watch events during the code execution.

To run the application, from the Run menu, select Resume, or alternatively click the Resume button in the toolbar.

## 6 SW toolchains helpful references and links

The following table provides useful references about integrated development environments described in this document:

**Table 2. Links to software toolchains**

Toolchain	Download link
EWARM	<a href="http://www.iar.com">www.iar.com</a>
MDK-ARM	<a href="http://www.keil.com">www.keil.com</a>
TrueSTUDIO	<a href="http://www.atollic.com">www.atollic.com</a>

## 7 Revision history

**Table 3. Document revision history**

<b>Date</b>	<b>Revision</b>	<b>Changes</b>
04-Aug-2015	1	Initial release.

**IMPORTANT NOTICE – PLEASE READ CAREFULLY**

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2015 STMicroelectronics – All rights reserved