
Getting started with STM32F411E Discovery software Development Tools

Introduction

This document describes the software environment required to build an application around the STM32F411E Discovery board (32F411EDISCOVERY), and provides some development recommendations.

It provides guidelines to novice user on how to build and run a sample application and allows them to create and build their own application.

This document:

- presents the Toolchains supporting the STM32 Families :
 - IAR Embedded Workbench® for ARM (EWARM) by IAR Systems,
 - Microcontroller Development Kit for ARM (MDK-ARM) by Keil™,
 - TrueSTUDIO® by Atollic;
- describes where to find the ST-LINK/V2 driver to be installed before starting coding on any Integrated Development Environment;
- describes step by step how to execute and debug an existing project with one of the previously presented toolchains;
- describes step by step how to create a new project with one of the toolchains from a dummy project included in the firmware package;
- provides helpful information on the advanced debugging capabilities of the Serial Wire Viewer (SWV) asynchronous trace supported by ST-LINK. This feature is especially useful when debugging certain tricky problems user may find during application development.

This user manual cannot cover all the topics relevant to software development environments, but it demonstrates the first basic steps necessary to get started with the compilers/debuggers and provides links to the required documents to fully understand every step.

Contents

1	Getting started	5
1.1	System requirements	5
1.2	IDEs supporting STM32 families	6
1.3	ST-LINK/V2 installation and development	6
1.4	Firmware package	7
2	Executing and debugging firmware using a software toolchain	9
2.1	EWARM toolchain	9
2.2	MDK-ARM toolchain	11
2.3	TrueSTUDIO toolchain	14
3	STM32F411E advanced debugging	17
4	Software toolchains helpful references and links	18
5	Revision history	19

List of tables

Table 1. Toolchains supporting STM32F411E Discovery. 6

Table 2. Links to software toolchains 18

Table 3. Document revision history 19

List of figures

Figure 1.	Hardware environment	5
Figure 2.	STM32F411E Discovery package content	7
Figure 3.	IAR Embedded Workbench IDE	9
Figure 4.	EWARM project successfully compiled.	10
Figure 5.	Download and Debug button	10
Figure 6.	IAR Embedded Workbench debugger screen	11
Figure 7.	IAR Go button	11
Figure 8.	uVision4 IDE	12
Figure 9.	MDK-ARM project successfully compiled	12
Figure 10.	Start/Stop Debug Session button	13
Figure 11.	MDK-ARM debugger screen.	13
Figure 12.	Run button	13
Figure 13.	TrueSTUDIO workspace launcher dialog box	14
Figure 14.	TrueSTUDIO® import source select dialog box	14
Figure 15.	TrueSTUDIO® import projects dialog box	15
Figure 16.	TrueSTUDIO® project successfully compiled.	15
Figure 17.	Figure 17.TrueSTUDIO debug window.	16

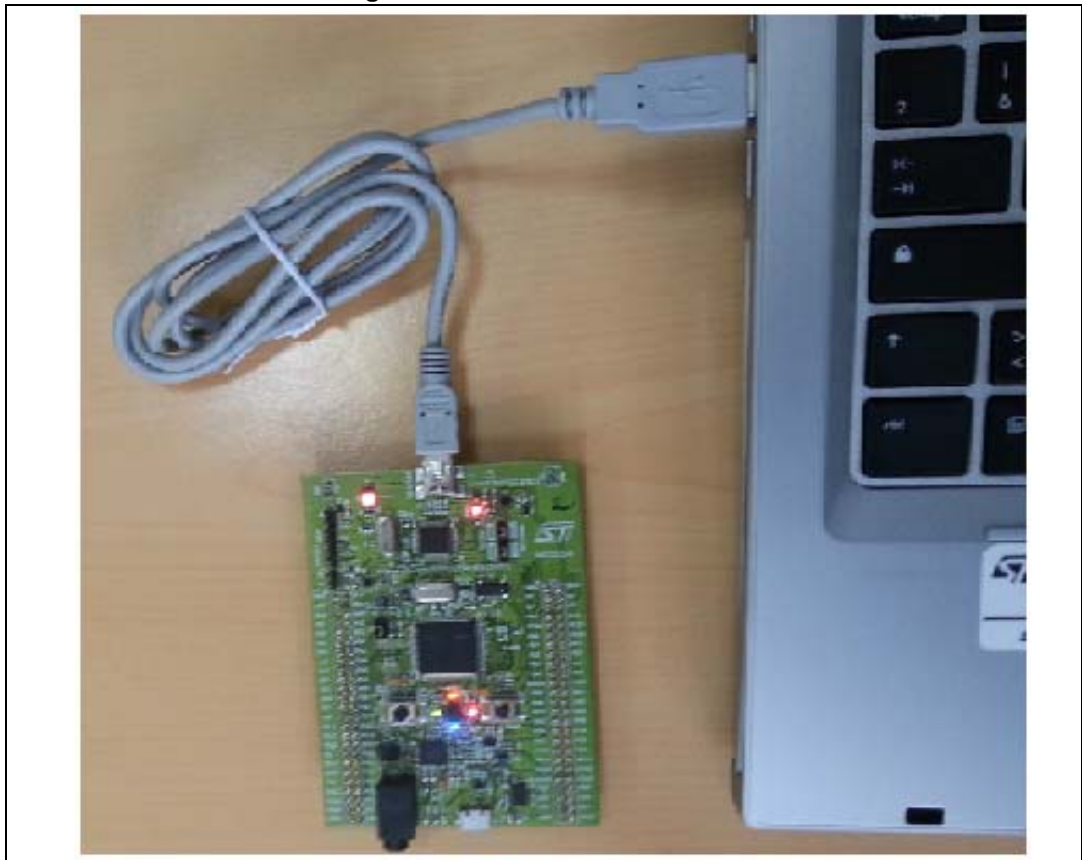
1 Getting started

1.1 System requirements

Before running user's application, please:

1. install user's preferred Integrated Development Environment (IDE),
2. install the ST-LINK V2 driver from the ST Website,
3. download the STM32F411E Discovery firmware from the ST Website,
4. establish the connection with the STM32F411E Discovery board as follows.

Figure 1. Hardware environment



The above steps will be detailed in the following sections.

To run and develop any firmware application on STM32F411E Discovery board, the minimum requirements are as follows:

- Windows (XP, 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 for debugging and programming

1.2 IDEs supporting STM32 families

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

The following table provides some general information about most used integrated development environments as well as the version supporting officially STM32F411E product.

Table 1. Toolchains supporting STM32F411E Discovery

Toolchain	Company	Version	Download link ⁽¹⁾
EWARM	IAR Systems®	7.20 and later	www.iar.com/en/Products/IAR-Embedded-Workbench/ARM/ – 30-day evaluation edition – KickStart edition(32Ko Limitation for Cortex M3/M4) – KickStart edition(16Ko Limitation for Cortex M0)
MDK-ARM	Keil™	5.xx and later with F4 Pack v 2.1.0	www.keil.com/demo/eval/arm.htm MDK-Lite (32Ko Code size limitation) http://www.keil.com/dd2/pack/#eula-container
TrueSTUDIO	© Atollic	5.2 and later	www.atollic.com/index.php/request-eval-license (*) – 32Ko Limitation (8Ko on Cortex-M0 and Cortex-M1) – 30 day Professional version (Trial)

1. Registration before download required.

1.3 ST-LINK/V2 installation and development

The STM32F411E 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 ST-LINK V2 and 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 PC's local hard disk.
 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_V2_USBdriver.exe
- RealView Microcontroller Development Kit (MDK-ARM)**
 The toolchain is installed by default in the C:\Keil directory on the PC's local hard disk; the installer creates a µVision4 shortcut in the start menu.
 When connecting the ST-LINK/V2 tool, the PC detects new hardware and prompts 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 PC's local hard disk.
The ST-Link_V2_USB.exe is installed automatically when installing the software toolchain.

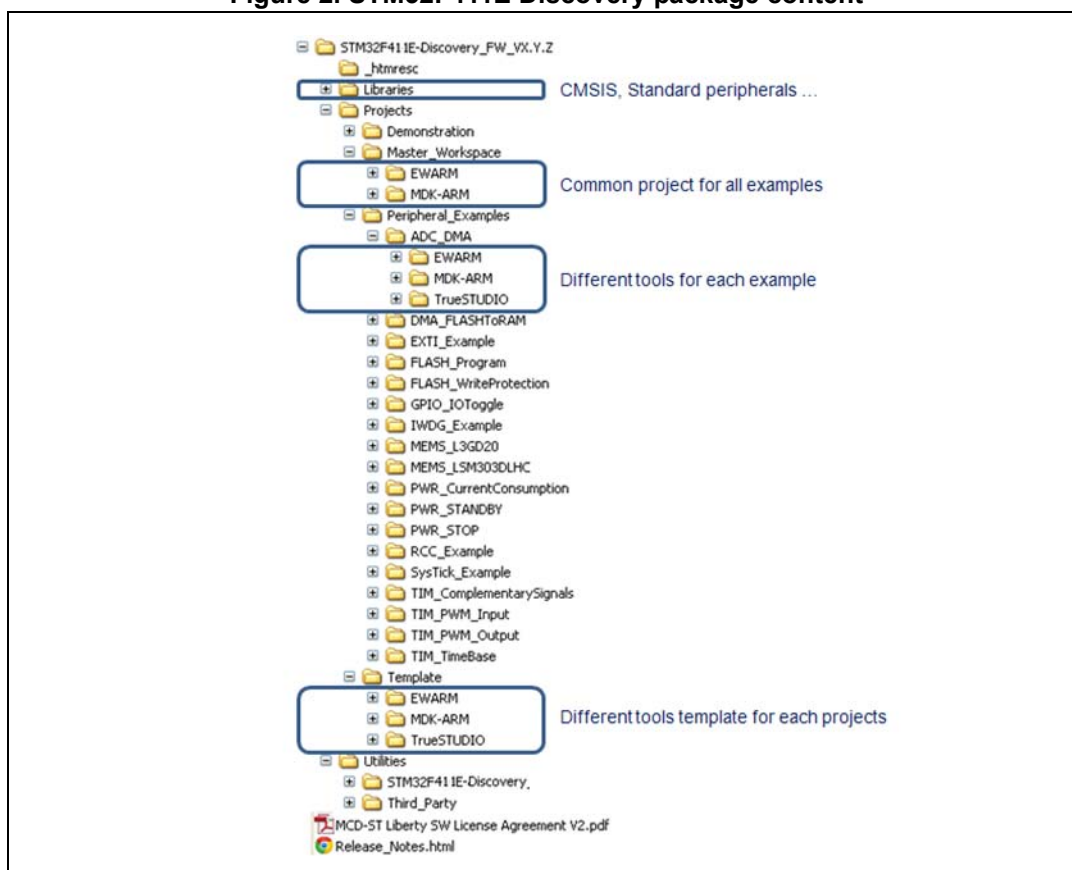
Complementary information on the firmware package content and the STM32F411E Discovery requirements is available in user manual UMxx Getting started with the STM32F411E Discovery kit.

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

1.4 Firmware package

The STM32F411E 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 “STM32F411E Discovery_FW_VX.Y.Z”, which contains the following subfolders:

Figure 2. STM32F411E Discovery package content



Template project: pre-configured project with empty main function to be customized by the user. This is helpful to start creating own application based on the peripherals drivers.

Master workspace: Collection of all projects available within this firmware package.

Peripheral examples: Including set of examples for each peripheral ready to be run.

2 Executing and debugging firmware using a software toolchain

This section describes how to compile/link and execute an existing project using the toolchains.

The steps below can be applied to an already existing example, demonstration or template project available at STM32F411E Discovery_FW_VX.Y.Z firmware available at www.st.com web site.

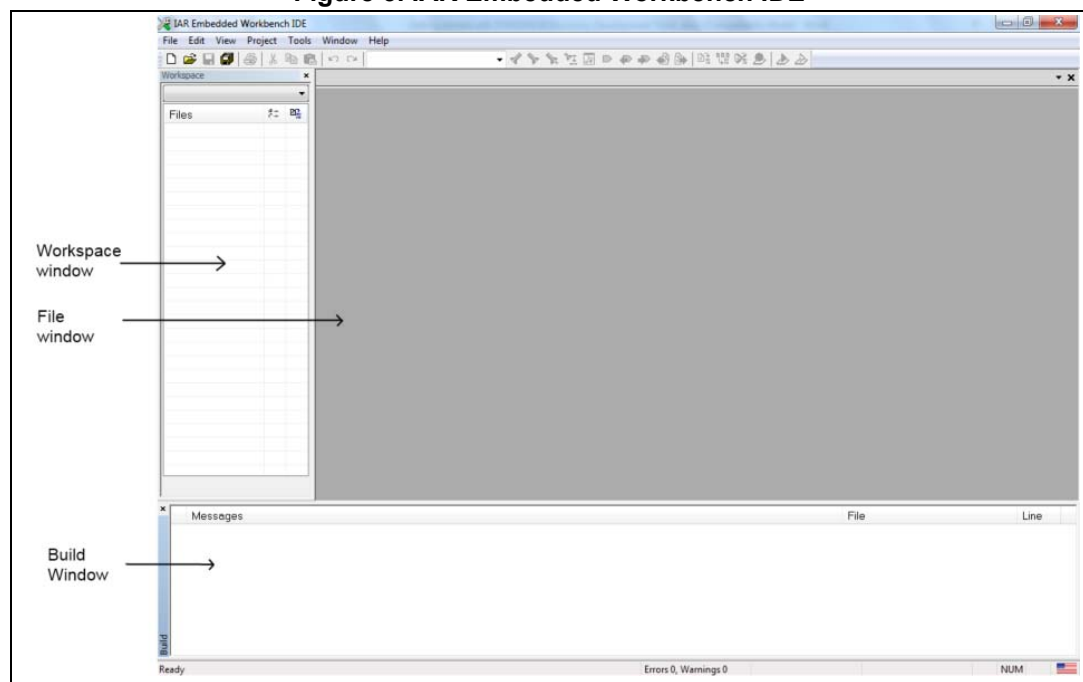
First of all, need to go through firmware/readme.txt file which contains a description of the firmware and hardware/software requirements.

2.1 EWARM toolchain

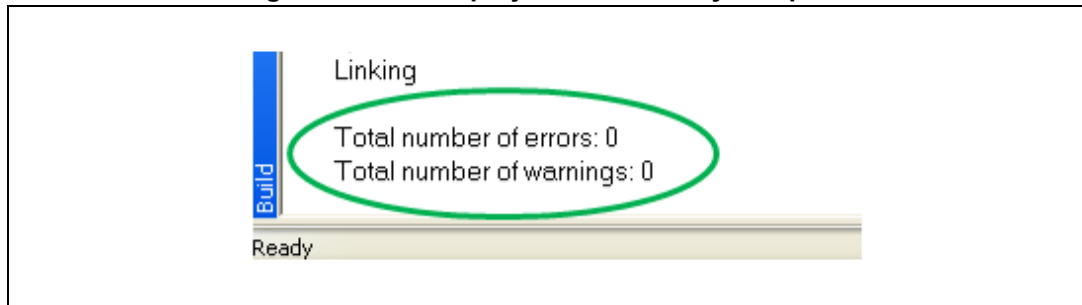
1. Open the IAR Embedded Workbench® for ARM (EWARM).

Figure 3 shows the basic names of the windows referred to in this document.

Figure 3. IAR Embedded Workbench IDE

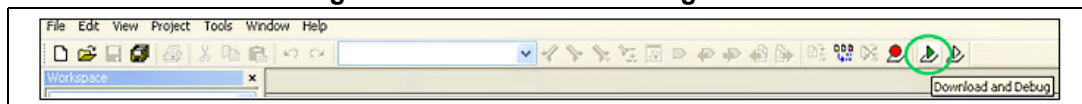


2. In the File menu, select Open and click Workspace to display the Open Workspace dialog box. Browse to select either an example or demonstration or template workspace file, and click Open to launch it in the Project window.
3. In the Project menu, select Rebuild All to compile the project.
4. If project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

Figure 4. EWARM project successfully compiled

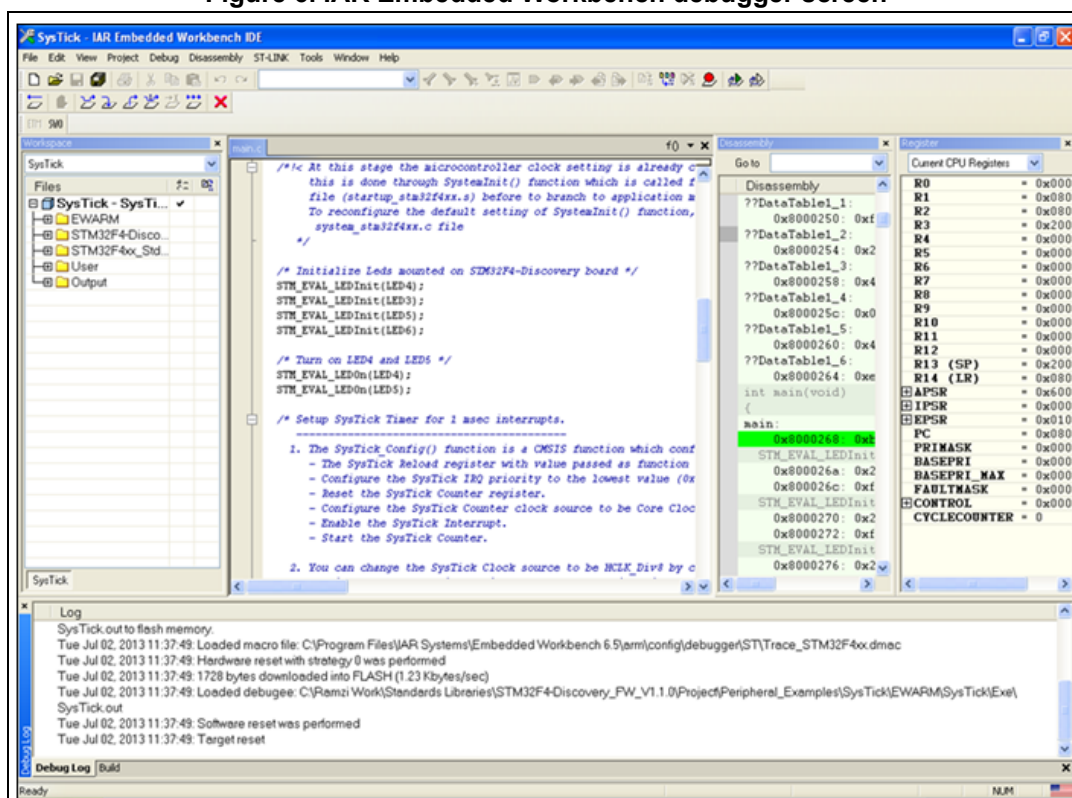
If need to change project settings (Include and preprocessor defines), simply go through the project options:

- 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

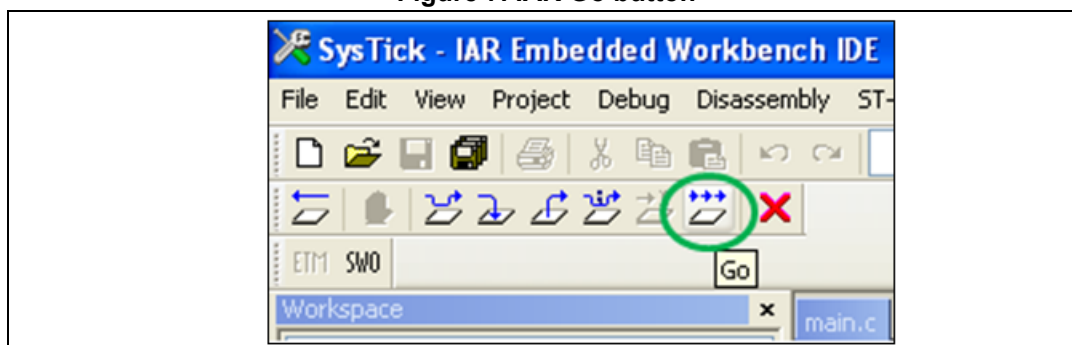
6. The debugger in the IAR Embedded Workbench can be used to debug source code at C and assembly levels, to set breakpoints, to monitor individual variables and watch events during the code execution.

Figure 6. IAR Embedded Workbench debugger screen



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

Figure 7. IAR Go button

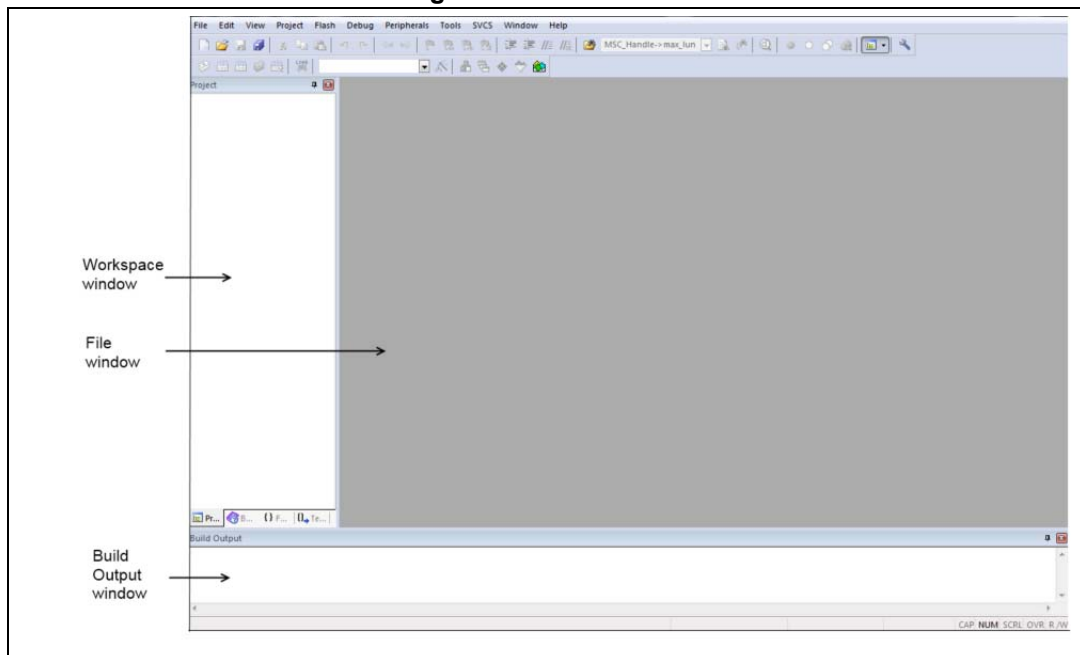


2.2 MDK-ARM toolchain

1. Open the Keil MDK-ARM Microcontroller Kit.

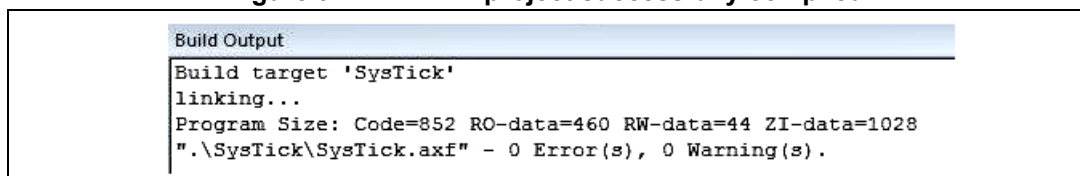
Figure 8 shows the basic names of the “Keil uVision4” windows referred to in this document.

Figure 8. uVision4 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 project
4. If project is successfully compiled, the following window in [Figure 9](#) is displayed.

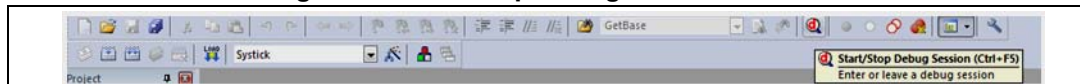
Figure 9. MDK-ARM project successfully compiled



If need to change project settings (Include and preprocessor defines), simply go through the project options:

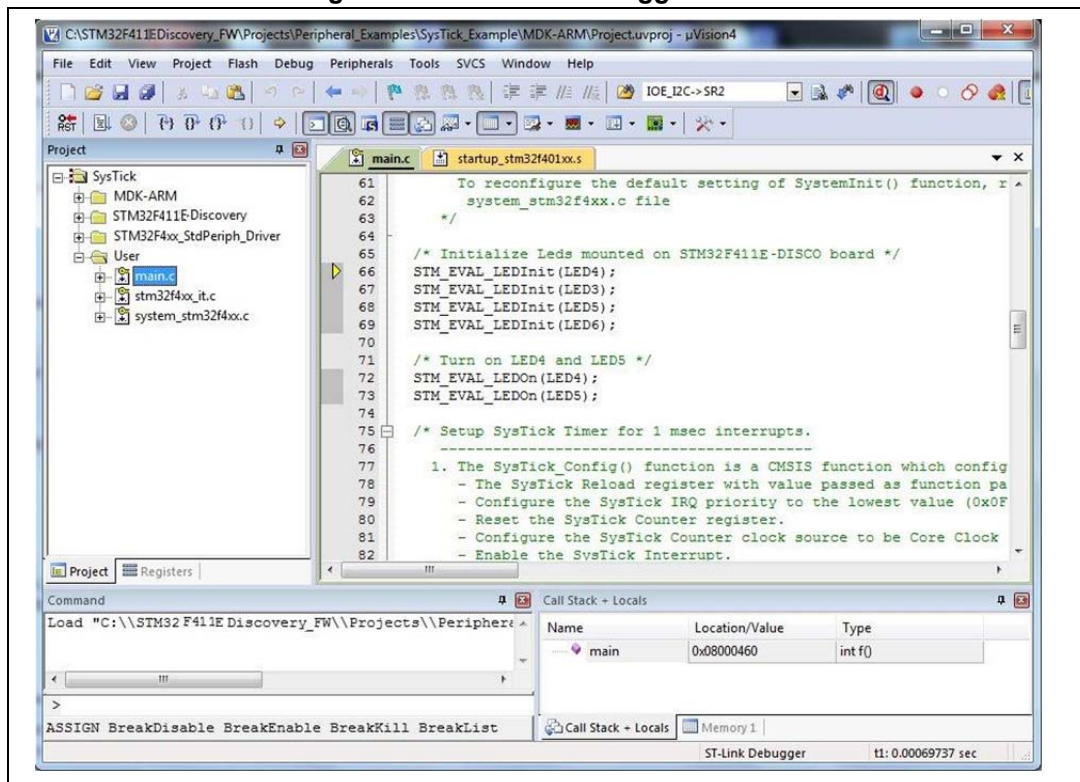
- For Include directories
Project>Options for Target > C/C++ > Include Paths
 - For pre-processor defines
Project>Options for Target > C/C++ > Preprocessor symbols > Define
5. In the MDK-ARM IDE, from the Debug menu, select Start/Stop Debug Session or, alternatively, click the Start/Stop Debug Session 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 user's application, from the Debug menu, select Run. Alternatively, click the Run button in the toolbar to run application.

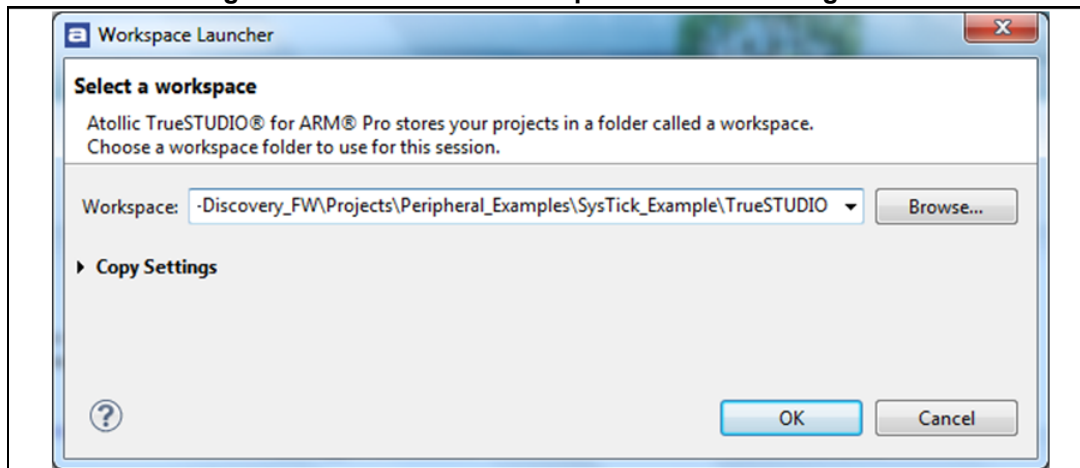
Figure 12. Run button



2.3 TrueSTUDIO toolchain

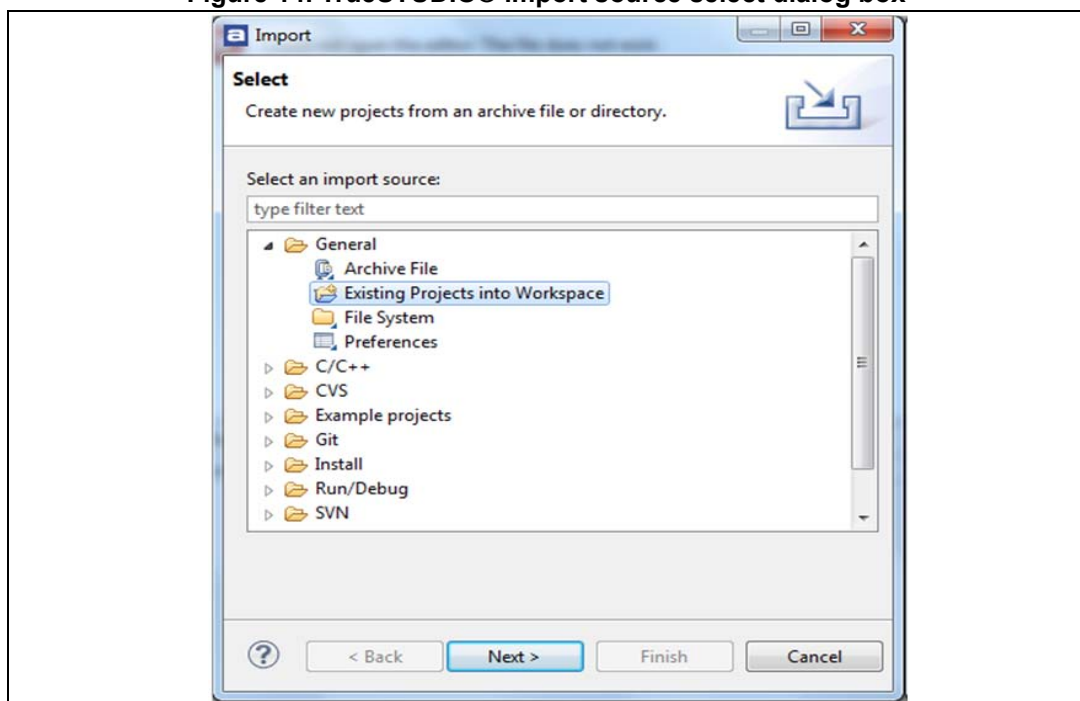
1. Open the Atollic TrueSTUDIO® for ARM product. The program launches and prompts for the Workspace location.

Figure 13. TrueSTUDIO workspace launcher dialog box



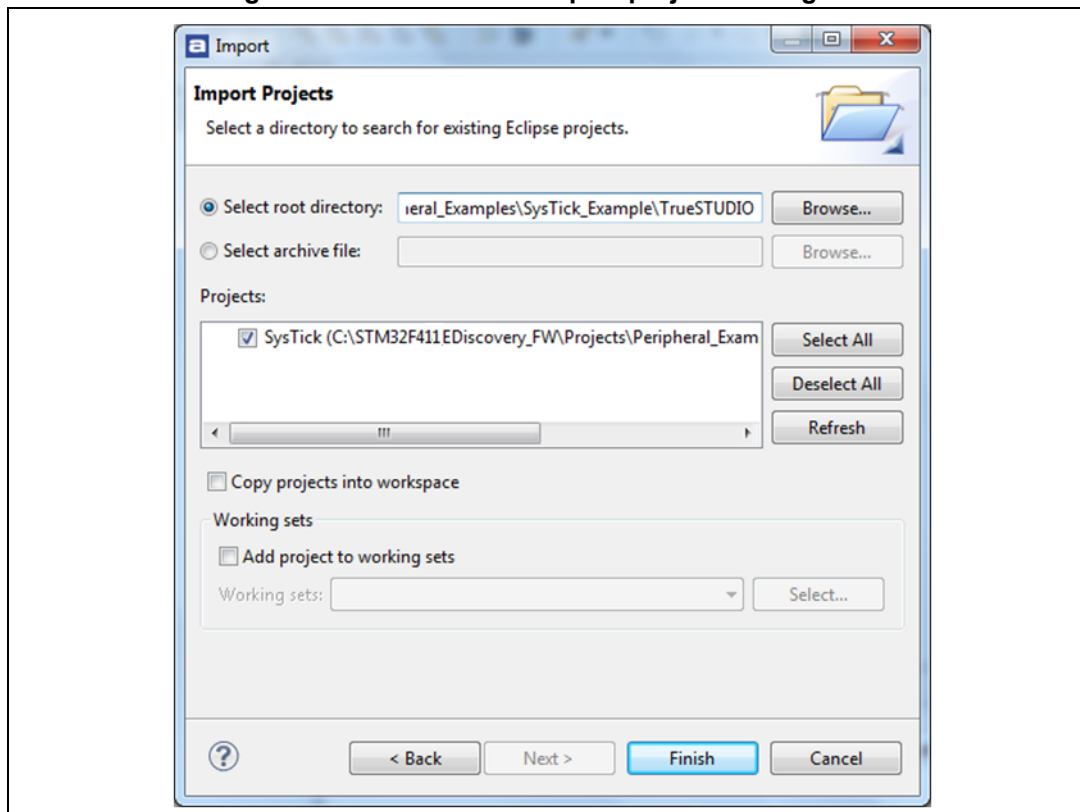
2. Browse to select a TrueSTUDIO workspace of either an example or demonstration or template workspace file and click OK to load it.
3. To load an existing project in the selected workspace, select Import from the File menu to display the Import dialog box.
4. In the Import window, open General, select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

Figure 14. TrueSTUDIO® import source select dialog box



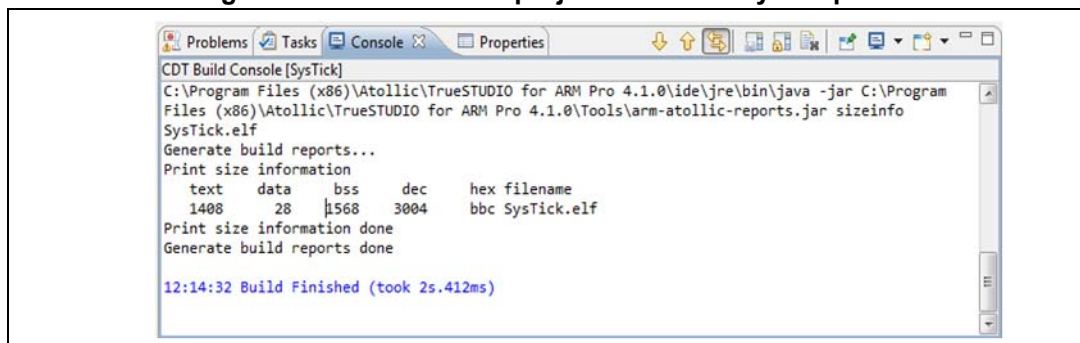
5. Click Select root directory, browse to the TrueSTUDIO workspace folder.

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 project is successfully compiled, the following messages display on the Console window.

Figure 16. TrueSTUDIO® project successfully compiled

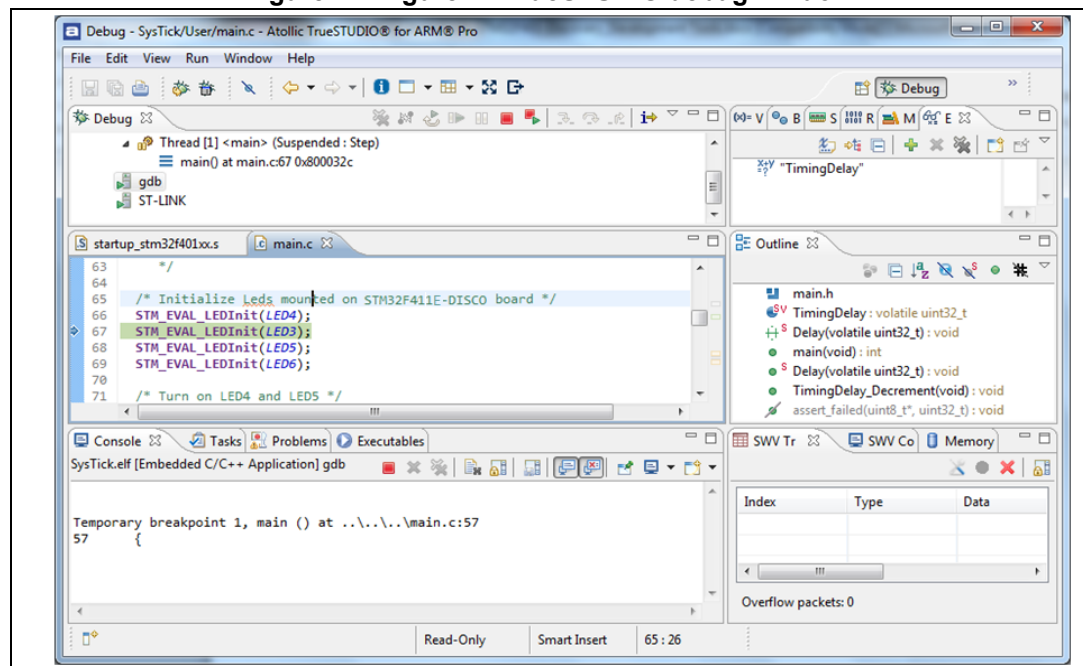


If needs to change the project settings (Include directories and preprocessor defines), simply go through Project>Properties, select C/C++ Build>Settings from the left panel:

- 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 11](#))

Figure 17. 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 user's application, from the Run menu, select Resume, or alternatively click the Resume button in the toolbar.

3 STM32F411E advanced debugging

The STM32 family using the Cortex-M4 processor has many interrupts and it can be difficult to determine when they are being activated and how often.

The serial Wire Viewer (SWV) on the STM32F411E family makes this task easy. SWV displays PC samples, exceptions (including interrupts), data reads and writes, ITM (printf), CPU counters and a timestamp. This information comes from the ARM CoreSight™ debug module integrated into STM32F411E CPU.

SWV does not steal any CPU cycles and is non-intrusive (Except for the ITM Debug printf Viewer).

The SWV configuration on the template project allows:

1. Retargeting printf to ITM stimulus port(0). This facilitates the display of the debug messages)
How to use it:
 - EWARM: View > terminal IO
 - MDK-ARM: View > Serial Windows Debug (printf) Viewer
 - TrueSTUDIO: View > SWV Console
2. Exception trace:
 - Provides information on the exception:
 - Entry: when the exception enters.
 - Exit: When it exits or returns.
 - Return: When all the exceptions have returned to the main
 - How to use it:
 - EWARM: ST-LINK>Interrupt log
 - MDK-ARM: View > Trace > Exceptions
 - TrueSTUDIO: View > SWV Exception Trace Log
3. Function profiler: shows timing information for the functions in an application
 - EWARM: ST-LINK>Function Profiler
 - MDK-ARM: View > Analysis Window > Code Coverage
 - TrueSTUDIO: View > SWV Statistical Profiling
4. Data Trace Timeline: shows a graphical representation of the data
 - EWARM: ST-LINK>Timeline (Data log)
 - MDK-ARM: View > Analysis Window > Logic Analyzer
 - TrueSTUDIO: View > SWV Data Trace Timeline

4 Software 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	www.iar.com/en/Products/IAR-Embedded-Workbench/ARM/- EWARM_UserGuide
MDK-ARM	www.keil.com/demo/eval/arm.htm www.keil.com/arm/mdk.asp
TrueSTUDIO	www.atollic.com/index.php/request-eval-license

5 Revision history

Table 3. Document revision history

Date	Revision	Changes
09-Jun-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