UM1790
User manual

Getting started with STM32L053 discovery kit software development tools

Introduction

This document describes the software environment recommendations required to build an application using the STM32L053 discovery kit (32L0538DISCOVERY).

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

The first chapter presents 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 first basic steps necessary to get started with the compilers/debuggers and includes references for complementary information.

Contents

List of tables

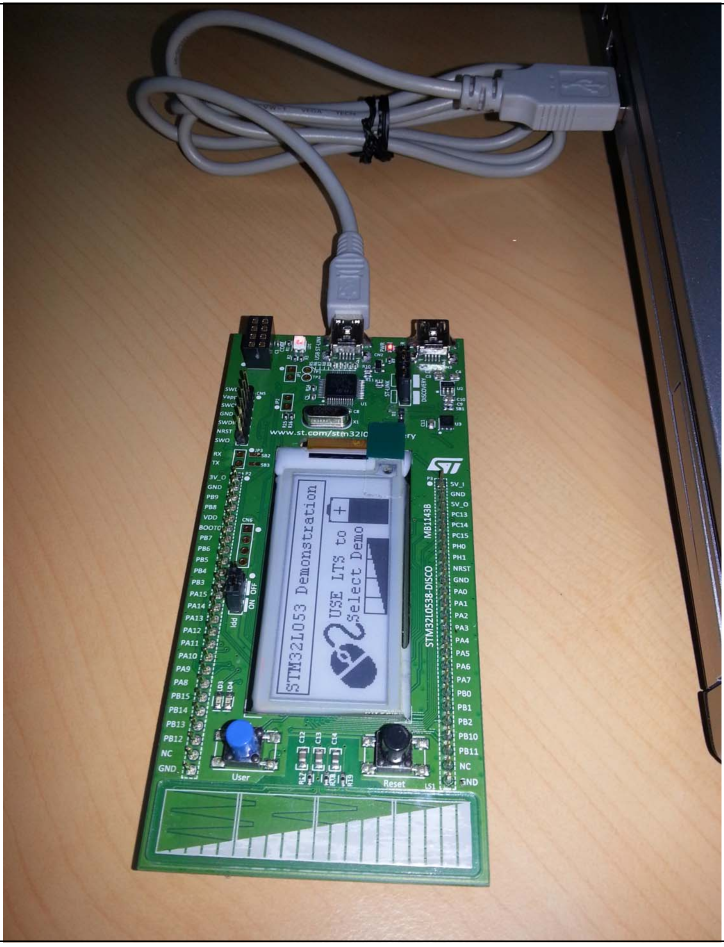
List of figures

1 System requirements

Before running your application, you should:

1. Install your preferred Integrated Development Environment (IDE)
2. ST-LINK/V2-1 driver will be installed automatically. In case of problem, you can proceed with manual installation of the driver from toolchains install directory (further details are available in Section 2).
3. Download the STM32L0538 discovery kit firmware from www.st.com/stm32l0-discovery.
4. Establish the connection with the STM32L0538 discovery board as following:

Figure 1. Hardware environment



The above steps will be details in the coming sections.

To run and develop any firmware applications on your STM32L053 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. 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 partner), completed with innovative tools from STMicroelectronics.

The following table regroups general information about most used integrated development environments as well as the version supporting officially STM32L053 product.

Table 1. Useful links

Toolchain	Company	Compiler	Version	Download link[1]
EWARM	IAR Systems®	IAR C/C++	7.10 and later	www.iar.com: – 30-day evaluation edition – KickStart edition(16Ko Limitation for Cortex® M0)
MDK-ARM[2]	Keil®	ARMCC	5.01 and later	www.keil.com: MDK-Lite (32Ko Code size limitation)
TrueSTUDIO®	Atollic	GNUC	5.0.0 and later	www.atollic.com[1] – 32Ko Limitation (8Ko on Cortex®-M0 and Cortex®-M1) – 30 day Professional version (Trial)

1. Registration before download is required.
2. Device database is updated separately from MDK-ARM release.

3 ST-LINK/V2-1 installation

STM32L053 discovery board includes an ST-LINK/V2-1 embedded debug tool interface. The interface needs an ST-LINK/V2-1 dedicated USB driver to be installed. This driver is available at www.st.com searching for ST-LINK V2-1 and is supported within 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/V2-1 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 PC's local hard disk; the installer creates a start menu µVision4 shortcut.

  When connecting the ST-LINK/V2-1 tool, the PC detects new hardware and asks to install the ST-LINK_V2_USB driver. The "found new hardware wizard" appears and guides you through the steps needed 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.

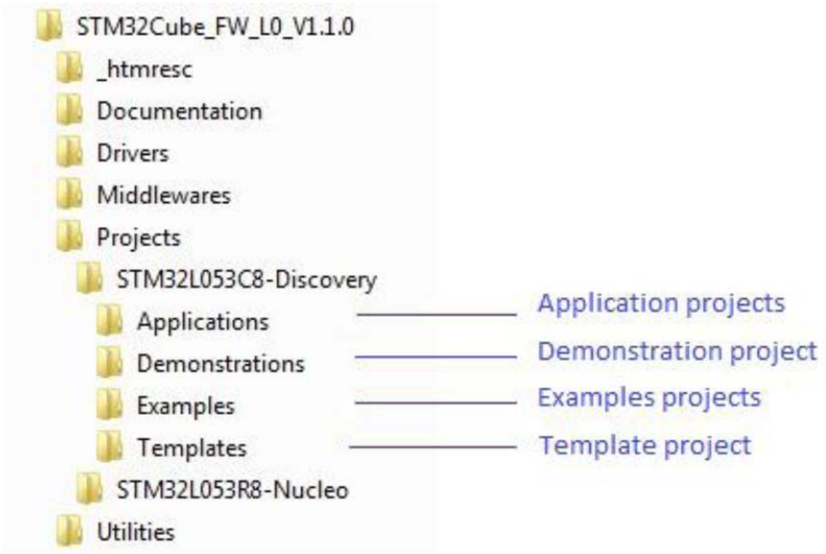
Complementary information on the firmware package content and the STM32L0538 discovery kit requirements are available on the Getting started with STM32Firmware.

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

4 Firmware package

The STM32L053 discovery kit firmware applications, demonstration and IPs examples are provided in one single package and supplied in one single zip file. The extraction of the zip file generates one folder, STM32L053-Discovery_FW_VX.Y.Z, which contains the following subfolders:

Figure 2. Package contents



Application examples: Including set of applications with pre-configured projects.

Demonstrations: Including set of demonstrations with pre-configured projects.

Peripheral examples: Including set of examples with pre-configured projects for each peripheral.

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

5 Executing and debugging firmware using software toolchains

5.1 EWARM toolchain

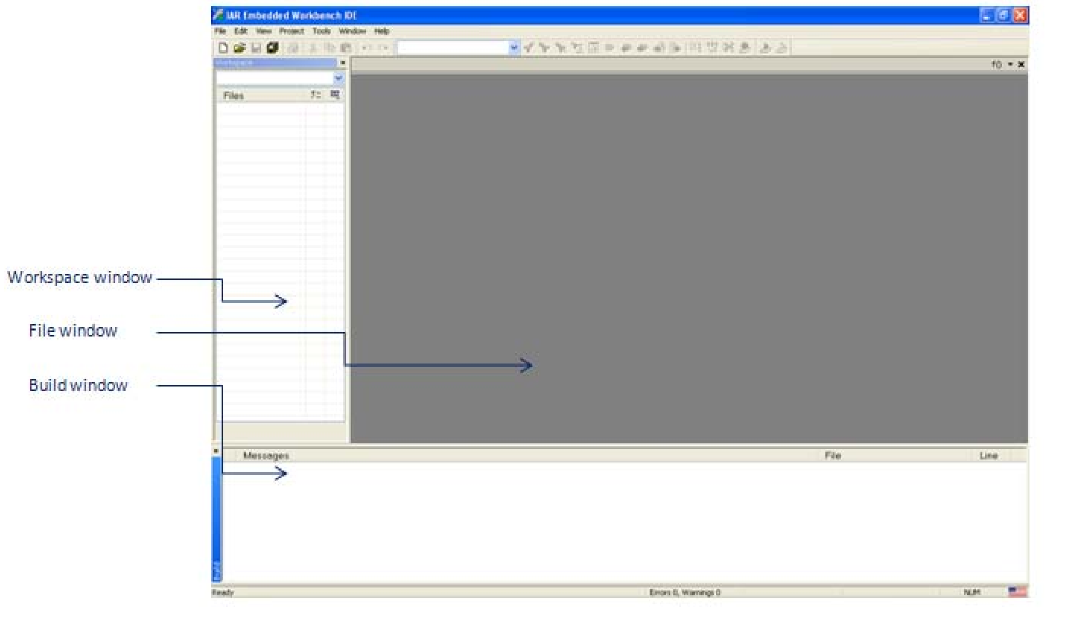
The following is the procedure for compiling/linking and executing an existing EWARM project.

Steps below can be applied to an already existing example, demonstration or template project included in STM32L053 discovery package that is available at www.st.com web site.

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

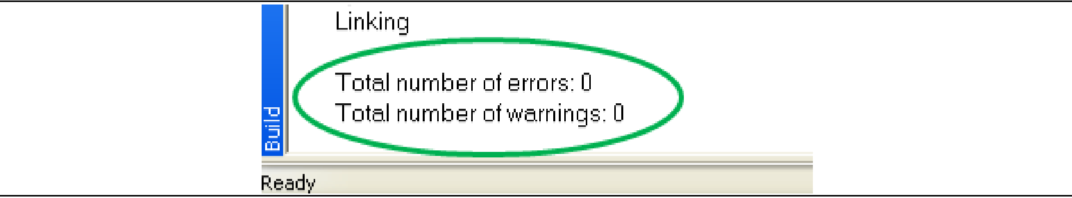
1. Open the IAR Embedded Workbench® for ARM (EWARM). *Figure 3*shows the basic names of the windows referred to this document.

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 your project
4. If your project is successfully compiled, the following window in *Figure 4* is displayed

Figure 4. EWARM project successfully compiled



Linking

Total number of errors: 0
Total number of warnings: 0

If you needs to change his project settings (Include and preprocessor defines), he has just to go through project options:

- For Include directories'

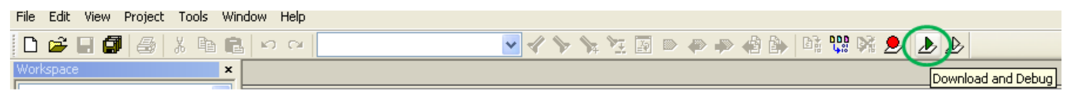
Project>Options…>C/C++ compiler>

- For pre-processor defines

Project>Options…C/C++ compiler>pre-processor>

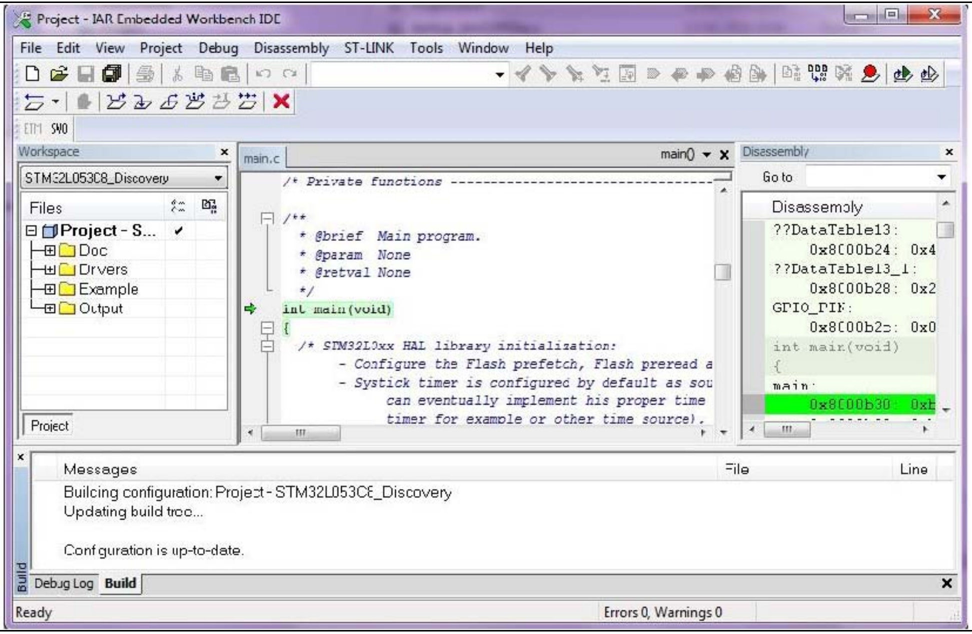
5. In the IAR Embedded Workbench IDE, from the **Project** menu, select **Download and Debug** or, alternatively, click the **Download and Debug** button the in toolbar, to program the Flash memory and begin debugging.

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, set breakpoints, monitor individual variables and watch events during the code execution.

Figure 6. IAR Embedded Workbench$^{®}$ debugger screen



```
/* Private functions -------------------------------

/**
  * @brief  Main program.
  * @param  None
  * @retval None
  */
int main(void)
{
  /* STM32L0xx HAL library initialization:
       - Configure the Flash prefetch, Flash preread a
       - Systick timer is configured by default as sou
             can eventually implement his proper time
             timer for example or other time source),
```

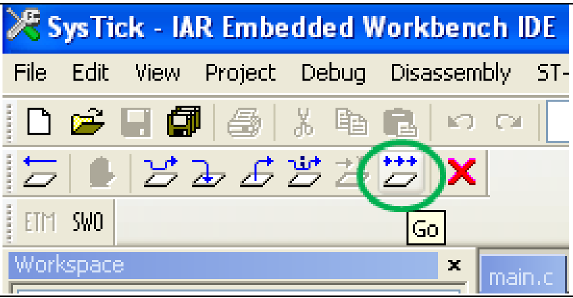
```
Disassembly
??DataTable13:
    0x8C00b24:  0x4
??DataTable13_1:
    0x8C00b28:  0x2
GPIO_PIN:
    0x8C00b2c:  0x0
int main(void)
{
main:
    0x8C00b30:  0xb
```

```
Messages                                              File      Line
Builcing configuration: Project - STM32L053C8_Discovery
Updating build tree...

Configuration is up-to-date.
```

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

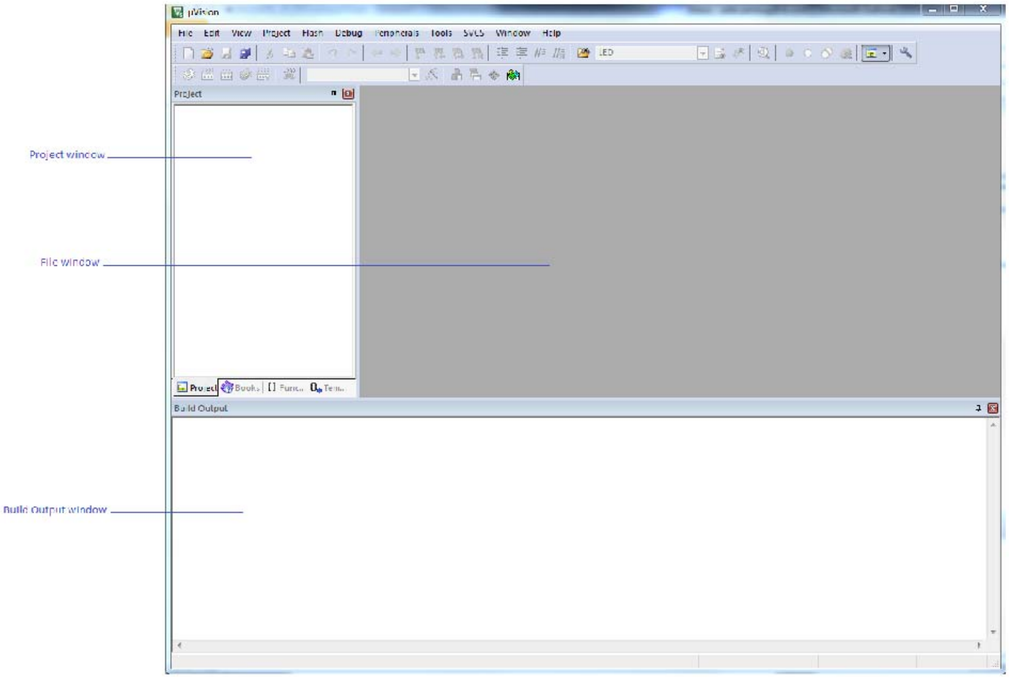
Figure 7. Go button



5.2 MDK-ARM toolchain

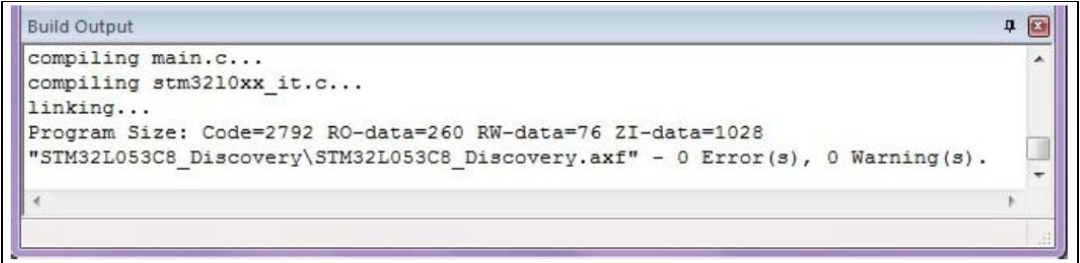
1. Open Keil MDK-ARM Microcontroller development kit, *Figure 8* shows the basic names of the “Keil uVision5” windows referred to in this document.

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 your project
4. If your project is successfully compiled, the following window in *Figure 9* is displayed

Figure 9. MDK-ARM project successfully compiled



```
compiling main.c...
compiling stm32l0xx_it.c...
linking...
Program Size: Code=2792 RO-data=260 RW-data=76 ZI-data=1028
"STM32L053C8_Discovery\STM32L053C8_Discovery.axf" - 0 Error(s), 0 Warning(s).
```

If you need to change your project settings (Include and preprocessor defines), you need just to go through project options:

- For include directories'

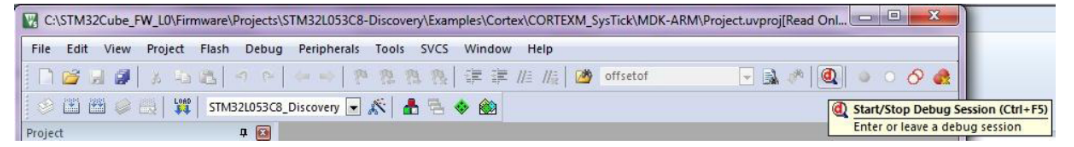
Project>Options for Target > C/C++ > Include Paths

- 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 the in toolbar, to program the Flash memory and begin debugging.

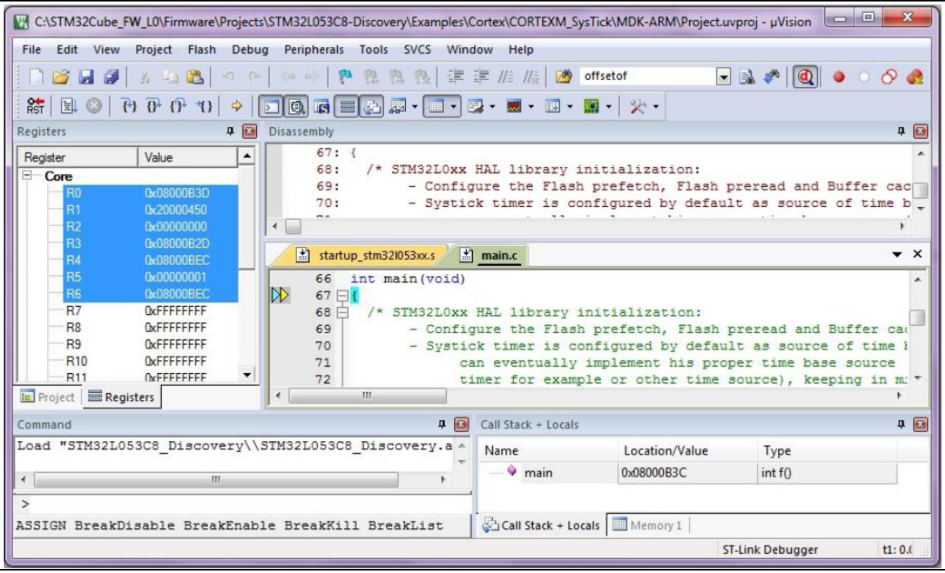
Figure 10. Start/Stop Debug Session button



C:\STM32Cube_FW_L0\Firmware\Projects\STM32L053C8-Discovery\Examples\Cortex\CORTEXM_SysTick\MDK-ARM\Project.uvproj[Read Onl...
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
offsetof
STM32L053C8_Discovery
Start/Stop Debug Session (Ctrl+F5)
Enter or leave a debug session
Project

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

Figure 11. MDK-ARM debugger screen


C:\STM32Cube_FW_L0\Firmware\Projects\STM32L053C8-Discovery\Examples\Cortex\CORTEXM_SysTick\MDK-ARM\Project.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
offsetof
RST
Registers
Disassembly
startup_stm32l053xx.s
main.c
Project
Registers
Command
Call Stack + Locals
Memory 1
ST-Link Debugger
t1: 0.(

Register	Value
Core	
R0	0x08000B3D
R1	0x20000450
R2	0x00000000
R3	0x08000B2D
R4	0x08000BEC
R5	0x00000001
R6	0x08000BEC
R7	0xFFFFFFFF
R8	0xFFFFFFFF
R9	0xFFFFFFFF
R10	0xFFFFFFFF
R11	0xFFFFFFFF

```
    67: {
    68:    /* STM32L0xx HAL library initialization:
    69:         - Configure the Flash prefetch, Flash preread and Buffer cac
    70:         - Systick timer is configured by default as source of time b
```

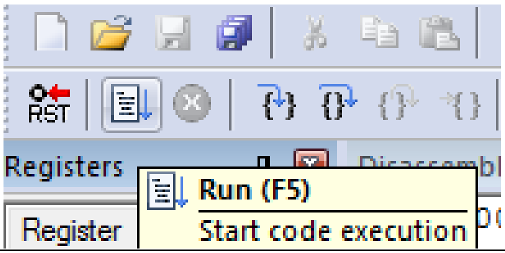
```
int main(void)
{
  /* STM32L0xx HAL library initialization:
       - Configure the Flash prefetch, Flash preread and Buffer cac
       - Systick timer is configured by default as source of time b
             can eventually implement his proper time base source
             timer for example or other time source), keeping in m:
```

```
Load "STM32L053C8_Discovery\\STM32L053C8_Discovery.a
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList
```

Name	Location/Value	Type
main	0x08000B3C	int f()

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

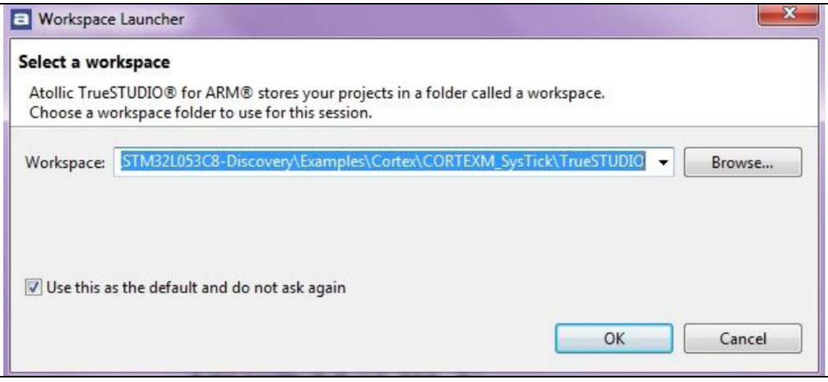
Figure 12. Run button



5.3 TrueSTUDIO® toolchain

1. Open Atollic TrueSTUDIO® for ARM product. The program launches and asks for the workspace location.

Figure 13. TrueSTUDIO® workspace launcher dialog box



Workspace Launcher

Select a workspace

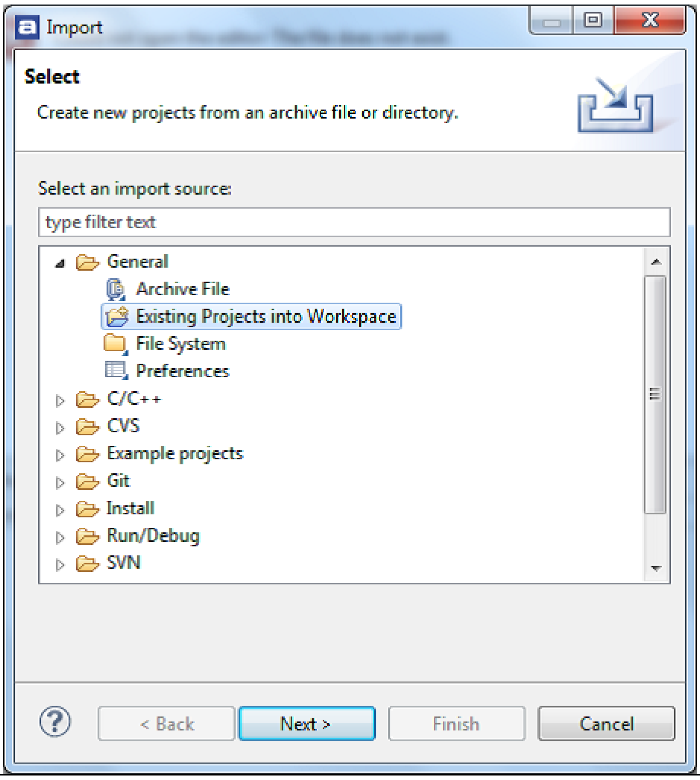
Atollic TrueSTUDIO® for ARM® stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: STM32L053C8-Discovery\Examples\Cortex\CORTEXM_SysTick\TrueSTUDIO

☑ Use this as the default and do not ask again

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. Atollic TrueSTUDIO® inport source select dialog box



Import

Select

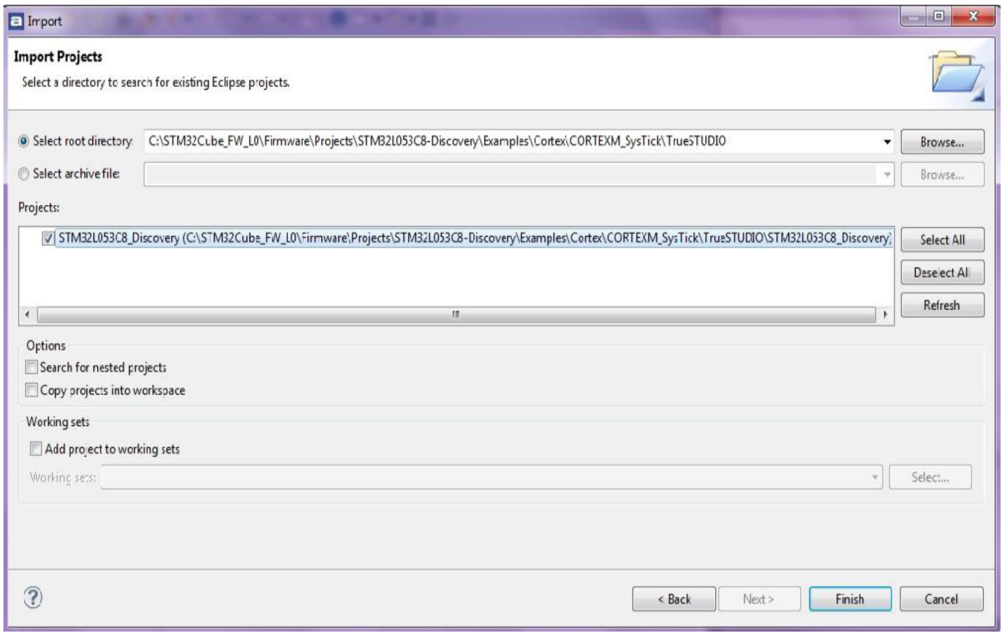
Create new projects from an archive file or directory.

Select an import source:

type filter text

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

Figure 15. Atollic TrueSTUDIO® import projects dialog box



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\STM32Cube_FW_L0\Firmware\Projects\STM32L053C8-Discovery\Examples\Cortex\CORTEXM_SysTick\TrueSTUDIO

○ Select archive file:

Projects:

☑ STM32L053C8_Discovery (C:\STM32Cube_FW_L0\Firmware\Projects\STM32L053C8-Discovery\Examples\Cortex\CORTEXM_SysTick\TrueSTUDIO\STM32L053C8_Discovery)

Options

□ Search for nested projects

□ Copy projects into workspace

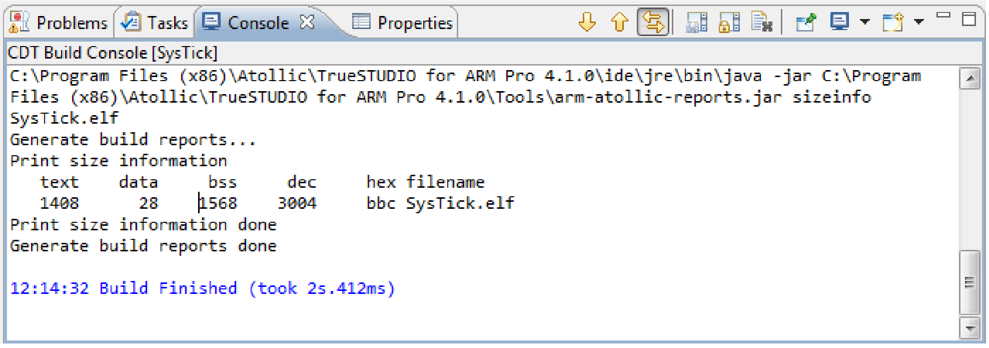
Working sets

□ Add project to working sets

Working sets:

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 your project is successfully compiled, the following messages will be displayed on the Console window.

Figure 16. TrueSTUDIO® project successfully compiled



```
CDT Build Console [SysTick]
C:\Program Files (x86)\Atollic\TrueSTUDIO for ARM Pro 4.1.0\ide\jre\bin\java -jar C:\Program
Files (x86)\Atollic\TrueSTUDIO for ARM Pro 4.1.0\Tools\arm-atollic-reports.jar sizeinfo
SysTick.elf
Generate build reports...
Print size information
   text    data     bss     dec     hex filename
   1408      28    1568    3004     bbc SysTick.elf
Print size information done
Generate build reports done

12:14:32 Build Finished (took 2s.412ms)
```

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

- For Include directories'

C Compiler>Directories>Include path

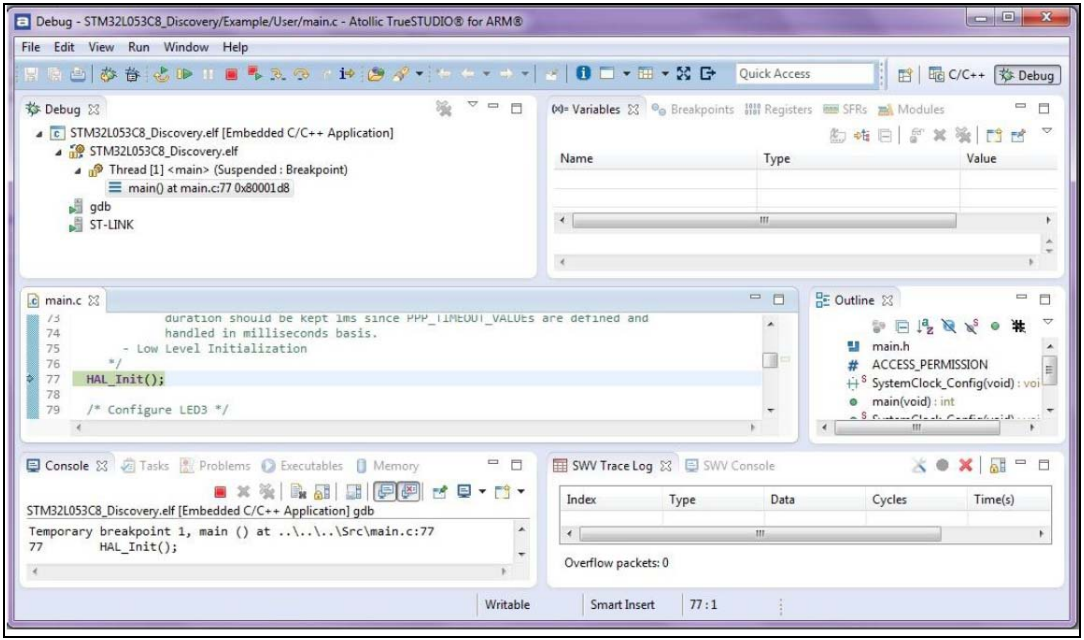
- For pre-processor defines

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 17*).

Figure 17. TrueSTUDIO® project successfully compiled


Debug - STM32L053C8_Discovery/Example/User/main.c - Atollic TrueSTUDIO® for ARM®
File Edit View Run Window Help
Quick Access
C/C++
Debug
Debug
STM32L053C8_Discovery.elf [Embedded C/C++ Application]
STM32L053C8_Discovery.elf
Thread [1] <main> (Suspended : Breakpoint)
main() at main.c:77 0x80001d8
gdb
ST-LINK
Variables
Breakpoints
Registers
SFRs
Modules
Name
Type
Value
main.c
Outline
main.h
ACCESS_PERMISSION
SystemClock_Config(void) : voi
main(void) : int
Console
Tasks
Problems
Executables
Memory
STM32L053C8_Discovery.elf [Embedded C/C++ Application] gdb
SWV Trace Log
SWV Console
Index
Type
Data
Cycles
Time(s)
Overflow packets: 0
Writable
Smart Insert
77 : 1

```
            duration should be kept 1ms since PPP_TIMEOUT_VALUEs are defined and
            handled in milliseconds basis.
        - Low Level Initialization
      */
  HAL_Init();

  /* Configure LED3 */
```

```
Temporary breakpoint 1, main () at ..\..\..\Src\main.c:77
77        HAL_Init();
```

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

To run your 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 regroups useful references about integrated development environments described in this document:

Table 2. References and links

Toolchain	Download link
EWARM	www.iar.com
MDK-ARM	www.keil.com
TrueSTUDIO®	www.atollic.com

7 Revision history

Table 3. Document revision history

Date	Revision	Changes
26-June-2014	1	initial release

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

ST PRODUCTS ARE NOT DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE ST PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER’S SOLE RISK, EVEN IF ST HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY ST AS BEING INTENDED FOR “AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL” INDUSTRY DOMAINS ACCORDING TO ST PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -
Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com