Introduction

The STM32 Discovery boards are low-cost and easy-to-use development platforms used to quickly evaluate and start a development with an STM32 device.

This document provides guidelines for beginners on STM32 MCU Discovery Kits on how to build and run a sample application and allows them to build and debug their application. It has the following structure:

- The first chapter presents software and hardware requirements (some toolchains supporting the STM32 families, ST-LINK/V2 installation and firmware package presentation).
- The second chapter provides a step by step guideline on how to build and debug an application using some toolchains:
  - IAR Embedded Workbench® for Arm® (EWARM) by IAR systems®
  - Microcontroller Development Kit for Arm® (MDK-ARM) by Keil®
  - TrueSTUDIO® by Atollic®
  - System Workbench for STM32 (SW4STM32) by AC6

This manual does not cover all the topics relevant to software development environments, but it describes the first basic steps necessary to get started with the compilers and debuggers. It also offers links to the documents needed to fully understand every single step.

<table>
<thead>
<tr>
<th>Type</th>
<th>Part numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>STM32 MCU Discovery Kits</td>
<td>STM32F7308-DK, 32F0308DISCOVERY, 32F072BDISCOVERY, 32F3348DISCOVERY, 32F411EDISCOVERY, 32F412GDISCOVERY, 32F413HDISCOVERY, 32F429iDISCOVERY, 32F469iDISCOVERY, 32F723EDISCOVERY, 32F746GDISCOVERY, 32F769iDISCOVERY, 32L0538DISCOVERY, 32L100CDISCOVERY, 32L152CDISCOVERY, 32L476GDISCOVERY, 32L496GDISCOVERY, 32L497iDISCOVERY, B-L072Z-LRWAN1, B-L475E-IOT01A, P-L496G-CELL01, P-L496G-CELL02, STM32F0DISCOVERY, STM32F3DISCOVERY, STM32F4DISCOVERY, STM32F4DISCOVERY, STM32F7508-DK, STM32VLDISCOVERY, STM32G071BDISCO</td>
</tr>
</tbody>
</table>
Contents

1 General information .......................................................... 5

2 System requirements ......................................................... 6
   2.1 IDEs supporting STM32 families ..................................... 7
   2.2 ST-LINK/V2 installation .............................................. 8
   2.3 Firmware package .................................................... 9

3 Executing and debugging firmware using the software toolchains .. 10
   3.1 EWARM toolchain .................................................... 10
   3.2 MDK-ARM toolchain .................................................. 12
   3.3 TrueSTUDIO toolchain ............................................... 13
   3.4 SW4STM32 toolchain ................................................. 17

4 Revision history .............................................................. 20
List of tables

Table 1. Applicable products ......................................................... 1
Table 2. Most used integrated development environments ..................... 7
Table 3. Document revision history .................................................. 20
List of figures

Figure 1. Hardware environment ....................................................... 6
Figure 2. Firmware package content .................................................. 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. Go button ........................................................................ 11
Figure 8. uVision5 IDE .................................................................... 12
Figure 9. MDK-ARM project successfully compiled ............................. 12
Figure 10. Start/Stop debug session button ......................................... 12
Figure 11. MDK-ARM debugger screen .............................................. 13
Figure 12. Run button ..................................................................... 13
Figure 13. TrueSTUDIO workspace launcher dialog box ....................... 13
Figure 14. Atollic TrueSTUDIO import source select dialog box .......... 14
Figure 15. Atollic TrueSTUDIO import projects dialog box .................. 14
Figure 16. TrueSTUDIO project successfully compiled ....................... 15
Figure 17. TrueSTUDIO debug window ............................................ 16
Figure 18. SW4STM32 workspace launcher dialog box ....................... 17
Figure 19. SW4STM32 import source select dialog box ....................... 17
Figure 20. SW4STM32 import projects dialog box ............................... 18
Figure 21. SW4STM32 project successfully compiled ........................... 18
Figure 22. SW4STM32 debug window ............................................... 19
1 General information

The STM32 Discovery boards embed STM32 32-bit microcontrollers based on the Arm®(a) Cortex®-M processor.

---

a. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and or elsewhere.
2 System requirements

Before running the application, the user must:
1. Install the preferred Integrated Development Environment (IDE)
2. The ST-LINK/V2 driver is installed automatically. In case of problem, the user can install it manually from the toolchains install directory (further details are available in Section 2.2: ST-LINK/V2 installation).

Note: This step is not needed for STM32VLDISCOVERY since it embeds an ST-LINK (not an ST-LINK/V2) which does not require a driver installation.


4. Establish the connection with the STM32 Discovery board as shown hereafter. In Figure 1, the STM32F072 Discovery board is chosen as an example.

Figure 1. Hardware environment
The steps numbered above are detailed in the coming sections.

The minimum requirements to run and develop any firmware application on the STM32 Discovery board are:

- Windows® OS (XP, 7, 8) or Linux 64-bit or Mac OS® X
- USB type A to Mini-B cable, used to power the STM32 Discovery board (through USB connector CN1) from the host PC and connect to the embedded ST-LINK/V2 for debugging and programming.

## 2.1 IDEs supporting STM32 families

The STM32 microcontrollers of 32-bit Arm Cortex-M core-based microcontrollers are supported by a complete range of software tools.

It encompasses traditional integrated-development environments such as IDEs with C/C++ compilers and debuggers from major third-party companies, free versions of up to 64 KB of code (depending on partner) and completed with innovative tools from STMicroelectronics.

The following table includes general information about some integrated development environments as well as the versions supporting the STM32 microcontrollers.

### Table 2. Most used integrated development environments

<table>
<thead>
<tr>
<th>Toolchain</th>
<th>Company</th>
<th>Compiler</th>
<th>Version</th>
<th>Information(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EWARM(2)</td>
<td>IAR Systems</td>
<td>IAR C/C++</td>
<td>7.60.1 and later</td>
<td><a href="http://www.iar.com">www.iar.com</a> – 30-day evaluation edition KickStart edition – 16 Ko limitation for Cortex M0</td>
</tr>
<tr>
<td>MDK-ARM(2)</td>
<td>Keil</td>
<td>ARMCC</td>
<td>5.18a and later</td>
<td><a href="http://www.keil.com">www.keil.com</a> – MDK-Lite – 32 Ko code size limitation</td>
</tr>
<tr>
<td>TrueSTUDIO(2)</td>
<td>Atolllic</td>
<td>GNUC</td>
<td>5.5.1 and later</td>
<td><a href="http://www.atolllic.com">www.atolllic.com</a> – 30 days professional version trial – 32 Ko limitation – 8 Ko on Cortex-M0 and Cortex-M1</td>
</tr>
<tr>
<td>SW4STM32(3)</td>
<td>AC6</td>
<td>GNUC</td>
<td>1.8 and later</td>
<td><a href="http://www.openstm32.org">www.openstm32.org</a> No limitation</td>
</tr>
</tbody>
</table>

1. Register prior to downloading the toolchain.
2. On Windows only.
3. The SW4STM32 does not support ST-LINK. Consequently, STM32VLDISCOVERY can not be used with SW4STM32 software toolchain.

Information on the toolchain version supporting the STM32 devices is available on the toolchain release note at the third-party website.
2.2 ST-LINK/V2 installation

All the STM32 Discovery boards include the ST-LINK/V2 embedded debug tool interface. This interface requires to install the ST-LINK/V2 dedicated USB driver. The STM32 Discovery boards include an ST-LINK/V2 embedded debug tool interface that is supported by the following 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 local hard disk.
  After installing the EWARM, the user should 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`.

- **RealView Microcontroller Development Kit for Arm (MDK-ARM) by Keil toolchain**
  The toolchain is installed by default in the `C:\Keil` directory on the PC local hard disk; the installer creates a start menu μVision5 shortcut.
  When connecting the ST-LINK/V2 tool, the PC detects the new hardware and asks to install the ST-LINK_V2_USB driver. The “Found New Hardware wizard” appears and guides the user 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 local hard disk.

- **AC6 SW4STM32 STM32**
  The toolchain is installed by default in the `C:\Program Files\AC6` directory on the PC local hard disk.
  The ST-Link_V2_USB.exe is installed automatically when installing the software toolchain.

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

The STM32 Discovery 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 the folder STM32-Discovery_FW_VX.Y.Z, which contains the subfolders shown in Figure 2 where the STM32 F072B-Discovery_FW_VX.Y.Z is chosen as an example.

Figure 2. Firmware package content

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

Example project includes the toolchain projects for each peripheral example ready to be run.

Applications includes set of applications ready to be run.

Demonstration includes the demonstration firmware ready to be run.
3 Executing and debugging firmware using the software toolchains

The user can follow the procedure described below to compile/link and execute an existing EWARM project. The following steps can be applied to an already existing example, demonstration or template project available at STM32 Discovery firmware package available on STMicroelectronics website. In this manual the STM32 F072B-Discovery_FW_VX.Y.Z is used as an example.

The user should go through the firmware/readme.txt file which contains the firmware description and hardware/software requirements.

3.1 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](image)

   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 is displayed.

   ![Figure 4. EWARM project successfully compiled](image)

   Figure 4. EWARM project successfully compiled
To change the project settings (Include and preprocessor defines), the user should go through project options:

- For 'Include directories', select Project>Options…>C/C++ compiler>
- For pre-processor, define 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 on the Download and Debug button in the tool bar, to program the Flash memory and start debugging.

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

To run the application, from the Debug menu, select Go. Alternatively, click on the Go button in the toolbar to run the application.
3.2 MDK-ARM toolchain

1. Open Keil MDK-ARM Microcontroller Kit,

   *Figure 8* shows the basic names of the "Keil uVision5" windows referred to in this document.

   *Figure 8. uVision5 IDE*

   ![uVision5 IDE](image)

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 window shown in *Figure 9* is displayed.

   *Figure 9. MDK-ARM project successfully compiled*

   ![MDK-ARM project successfully compiled](image)

If the user needs to change the project settings (Include and preprocessor defines), the user should go through the project options:

- For 'Include directories', select **Project>Options for Target > C/C++ > Include Paths**
- For pre-processor definition **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 tool bar to program the Flash memory and begin debugging.

   *Figure 10. Start/Stop debug session button*

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

### 3.3 TrueSTUDIO toolchain

1. Open Atollc TrueSTUDIO for Arm microcontrollers. The program launches and asks 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**.
5. Click **Select root directory**, browse to the TrueSTUDIO workspace folder.
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 will be displayed on the Console window.

![Figure 16. TrueSTUDIO project successfully compiled](image)

If the user needs to change the project settings (Include directories and preprocessor defines), 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, the user should select the project in the Project Explorer and press F11 to start a debug session (see Figure 17).
The debugger in the Atollc 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 the application, from the Run menu, select Resume, or alternatively click the Resume button in the toolbar.
3.4 SW4STM32 toolchain

1. Open the AC6 SW4STM32 for Arm microcontrollers. The program launches and prompts for the Workspace location.

   Figure 18. SW4STM32 workspace launcher dialog box

   ![Workspace Launcher Dialog Box]

2. Browse to select a SW4STM32 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 19. SW4STM32 import source select dialog box

   ![Import Source Select Dialog Box]

5. Click Select root directory, browse to the SW4STM32 workspace folder.
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.

If the user 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. See **Figure 22**.
The debugger in the AC6 SW4STM32 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.
4 Revision history

Table 3. Document revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>20-May-2016</td>
<td>1</td>
<td>Initial release.</td>
</tr>
<tr>
<td>25-Feb-2019</td>
<td>2</td>
<td>Added Table 1: Applicable products.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added Section 1: General information.</td>
</tr>
</tbody>
</table>
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.

© 2019 STMicroelectronics – All rights reserved