

Abstract

The latest version of this document is here: www.keil.com/apnotes/docs/apnt_323.asp

STM32CubeIDE is STMicroelectronics free toolchain for STM32 based microcontrollers. This application note shows how to create μ Vision projects based on device configurations made in STM32CubeIDE.

Prerequisites

- Arm Software:
 - o Arm Keil MDK v 5.28a or later. For SMT32 F0/L0/G0 devices please use free MDK edition available for these families.
 - o Device Family Pack for target device family.
- STM32 Software:
 - o STM32CubeIDE v1.0.2 or later.
 - o STM32CubeMX v5.3.0 or later.
 - o STM32Cube MCU package for target device.

The flow described in this application note is universal. To provide real-world references, the screenshots used throughout the document are based on an example application for a SMT32L4 device.

Contents

Using Keil MDK with STM32CubeIDE	1
Abstract	1
Prerequisites	1
Introduction	3
Software Setup	3
Device configuration in STM32CubeIDE	4
Code generation for Keil MDK	4
Using generated code in MDK	5
Summary	6
Useful links	6

Introduction

[STM32Cube](#) provides multiple software tools for developing applications on Arm Cortex-M based STM32 devices. [Arm Keil MDK](#) well integrates with this ecosystem. Developers can utilize the STM32Cube tools while additionally benefit from:

- the fast and intuitive [µVision IDE](#)
- the highly efficient [Arm C/C++ compiler](#)
- the advanced [µVision debug capabilities](#) such as
 - built-in RTOS-awareness
 - powerful trace support with code coverage and performance analysis
 - accurate power measurement
 - Event Recorder and Event Statistics for timing analysis and visibility into program operation

A [free MDK edition](#) for STM32 F0/L0/G0 devices is available at no cost for the end user, so that you can leverage all these benefits in your development project.

[STM32CubeMX](#) is a graphical configuration tool that simplifies configuration of the target device and generates C initialization code for the project. It relies on [STM32Cube MCU Packages](#) that contain device-specific descriptions, hardware abstraction layer (HAL) and a set of example applications. STM32CubeMX is available as a standalone tool and is also integrated into Eclipse-based [STM32CubeIDE](#).

Our documentation already [explains in details](#) how to use the powerful device configuration possibilities of STM32CubeMX with a Keil MDK project. Due to its convenience and full integration, this approach is recommended for use in new or existing MDK projects.

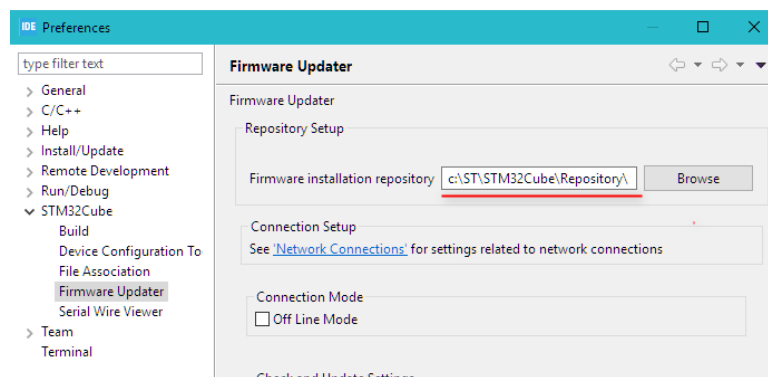
This application note additionally demonstrates the possibility of code generation for Keil MDK from configurations made in STM32CubeIDE.

Software Setup

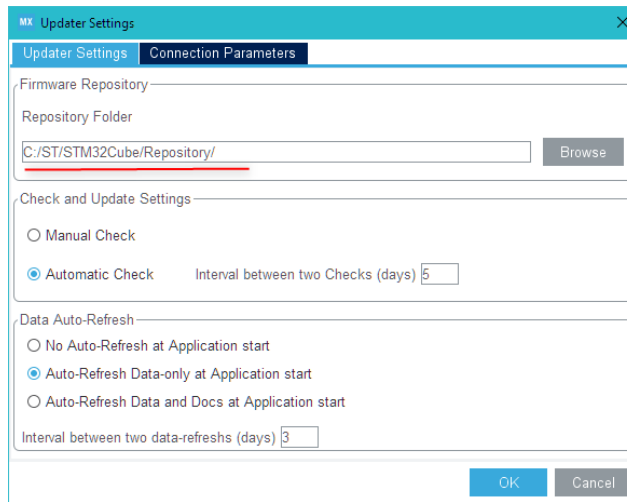
The STM32Cube integration with Keil MDK requires that a standalone version of STM32CubeMX is used. This enables code generation and project creation for Arm Keil MDK from STM32CubeIDE.

1. It is assumed that STM32CubeIDE is already installed on the PC and is open with an existing project in it (several example projects are typically available with the STM32Cube MCU packages).


Using menu **Window – Preferences – STM32Cube – Firmware Updater** observe the firmware installation repository used for STM32Cube MCU Packages



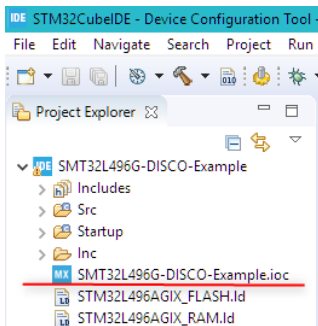
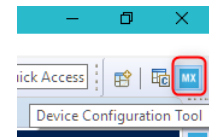
2. Download STM32CubeMX from www.st.com/stm32cubemx and install it on the PC. Start STM32CubeMX and go to **Help – Updater Settings..** to point to the same repository of as used by STM32CubeIDE. This ensures that same packages are used.




Device configuration in STM32CubeIDE

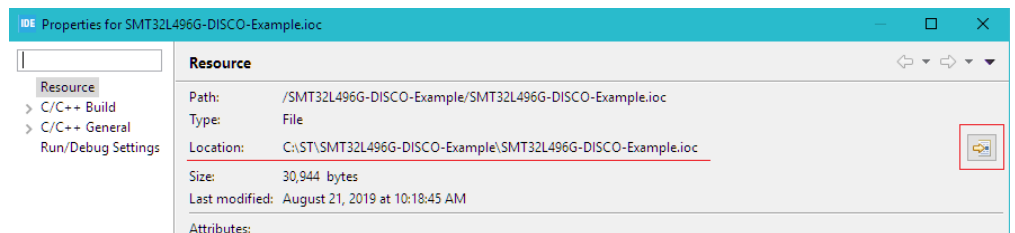
STM32CubeIDE contains a device configuration tool that allows users to graphically configure the target device. It is available as a perspective in Eclipse and can be accessed via the corresponding  button or using menu **Window – Perspective – Open Perspective – Other... – Device Configuration Tool**.

The configuration is kept in an **.ioc** file that is also visible in the Project Explorer.



STM32CubeMX uses this **.ioc** file to generate the code for the Keil MDK.

To locate the **.ioc** file, right-click on it and select **Properties** in the context menu. The file location is shown and you can open the containing directory in Windows Explorer using  button.

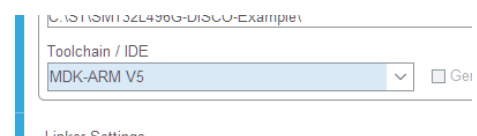


Code generation for Keil MDK

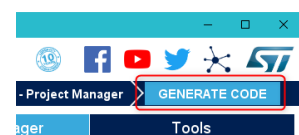
In Windows Explorer, double-click on the **.ioc** file to open it in the standalone STM32CubeMX application. The dialog offering migration to a newer STM32 MCU Package (Firmware Package) may appear. It is up to you to decide which version to use.

You can use the STM32CubeMX to modify the device configuration if required, but most importantly it allows project and code generation for Keil MDK.

Switch to the **Project Manager** tab. In the **Project** options, select MDK-ARM V5 as **Toolchain/IDE**.



Use the **GENERATE CODE** button in the upper-right corner to create a μ Vision project files in the MDK-ARM/ folder and to generate the code configuring the device.

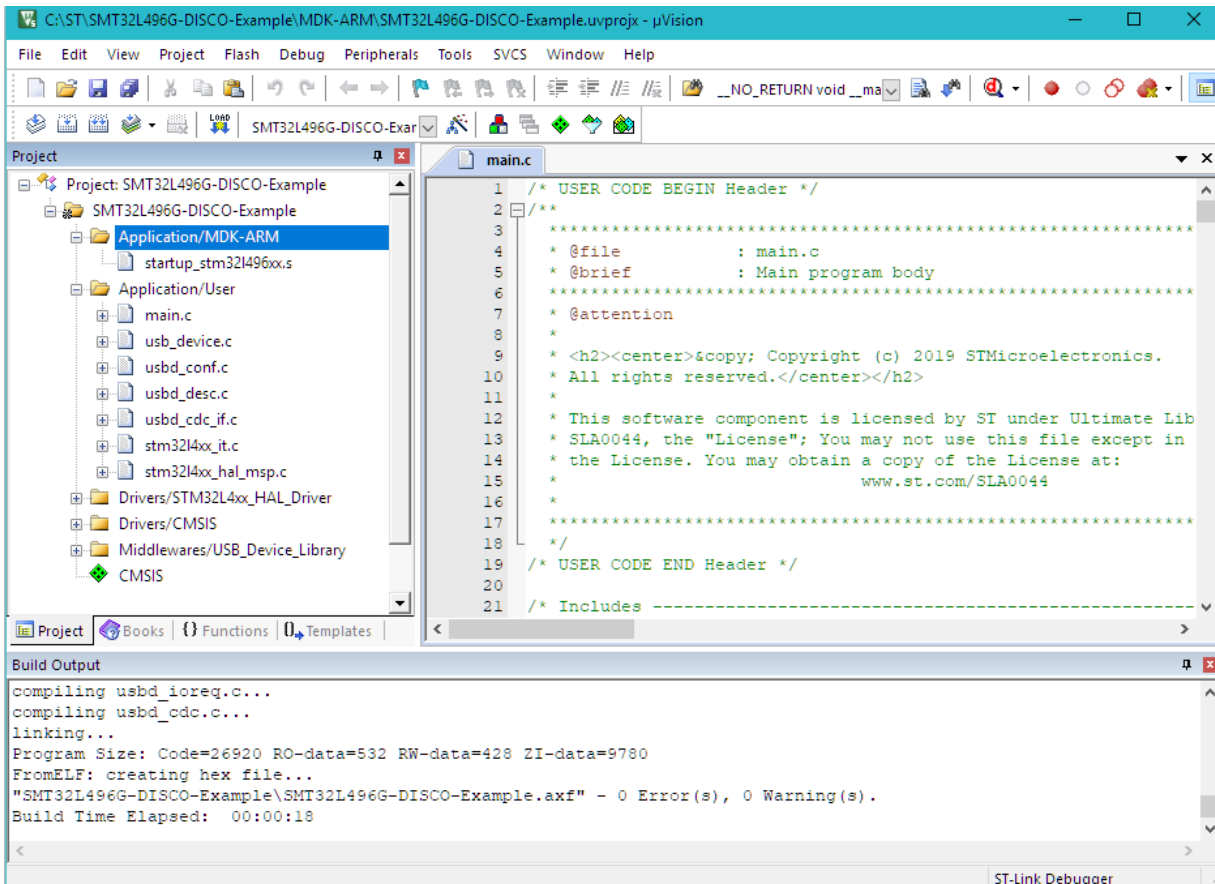


Once code generation is completed, you will be offered to open the μ Vision project.

The μ Vision project file is available in newly created MDK-ARM/ folder in the original STM32CubeIDE project directory.

Using generated code in MDK

Open the generated project in μ Vision IDE to browse through the project files and to edit them.



Note that only files in *Application/MDK-ARM* category are specific for Keil MDK, others are shared with the STM32CubeIDE project.

The generated project doesn't use any code from Device Family Packs (DFP) or CMSIS and can be compiled as is. This however makes project management less convenient comparing to the [development flow](#) based on standalone STM32CubeMX.

If compilation time is too long, go to **Options for Target** (🔧) dialog – **Output** tab and uncheck **Browse Information**.

For debugging the program in μ Vision, you still need to install the DFP for the target STM32 family as it contains necessary flash programming algorithms. This can be done via [Pack Installer](#) (📦).

Once the DFP is installed, you can load the compiled firmware to the device using Load button (📶), or start a debug session (🔍).

Summary

This application note explained the integration between Keil MDK and STM32Cube. It demonstrated how to generate code for Keil MDK using device configuration done in STM32CubeIDE.

Useful links

- [Using Keil MDK with STM32Cube](#) explains the recommended workflow for creating projects that utilize STM32Cube together with Keil MDK.
- [Free MDK STM32 F0/L0/G0 edition](#) special free MDK edition that supports STM32 Cortex-M0/M0+ devices without code size limitation.
- [Keil MDK for STM32](#) gives an overview of MDK resources specific to STM32.