Building Mbed OS projects with MDK-Community

KAN 340, March 2022, V 1.0

keil-feedback@arm.com

Abstract

Arm Mbed OS is a free, open-source IoT operating system with connectivity, security, storage, device management and machine learning. It includes all the features you need to develop a connected product based on an Arm Cortex-M microcontroller, including security, connectivity, an RTOS, and drivers for sensors and I/O devices.

Mbed OS itself is not available as an CMSIS-Pack but has its own configuration and dependency management system. This application note shows how to get started building Mbed OS projects with the MDK-Community edition (or any other MDK edition) by migrating projects from the Mbed Online Compiler to MDK.

Contents

Introduction

While Mbed OS is very popular amongst hobbyists and makers, historically it was not easy to use it with MDK due to different build systems and the lack of a completely free MDK edition. We have now introduced the MDK-Community edition that is covering all Cortex-M devices and does not impose any code size limit. It is free-to-use for hobbyists, makers, students, and academics for non-commercial use.

This application note explains how to export your Mbed OS (5 and above) projects to Keil MDK.

Prerequisites

You need to have the following software installed on your machine:

- Keil MDK – get the latest version from here: keil.com/demo/eval/arm.htm.
- A valid MDK license, such as MDK-Community. Any other (paid) license will do as well.

You also need:

- An Mbed-enabled development board.

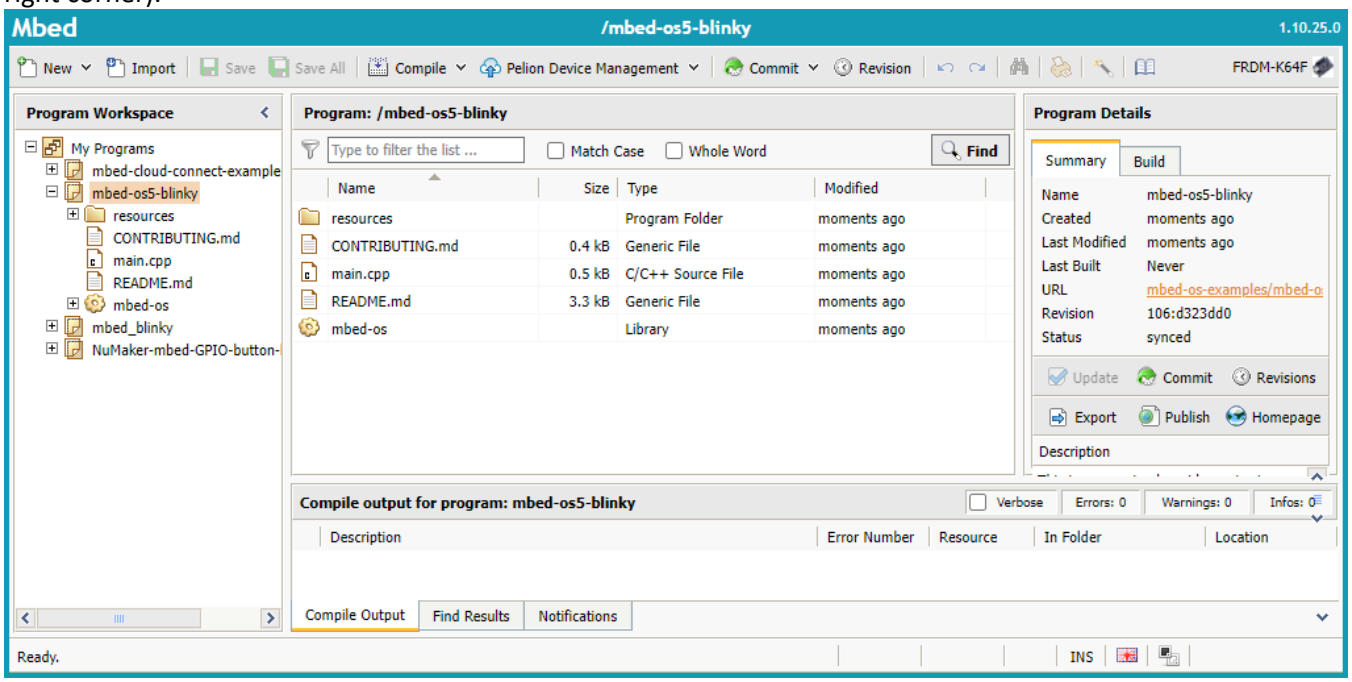
Note

The following instructions only work for Mbed OS 5.x and 6.x projects. Older Mbed 2 projects are not supported. Consider upgrading them first.

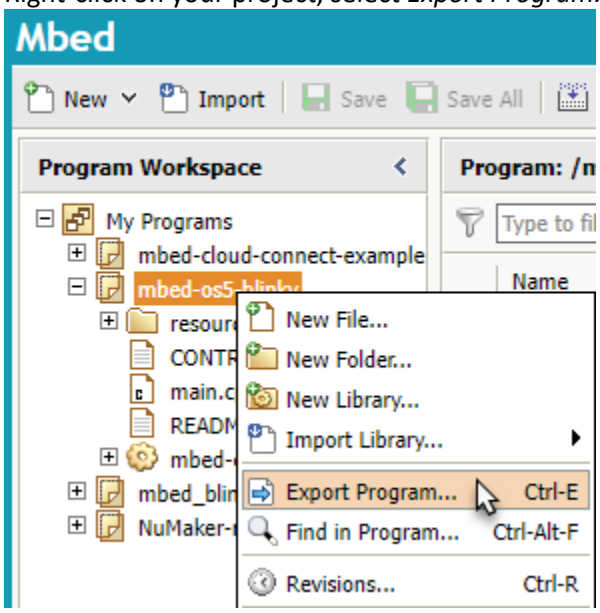
Export an existing project from the Mbed Online Compiler

The Mbed Online Compiler is a hosted web application that allows you to write and build code without requiring a local IDE or device drivers. Projects in the Online Compiler can be exported easily to µVision.

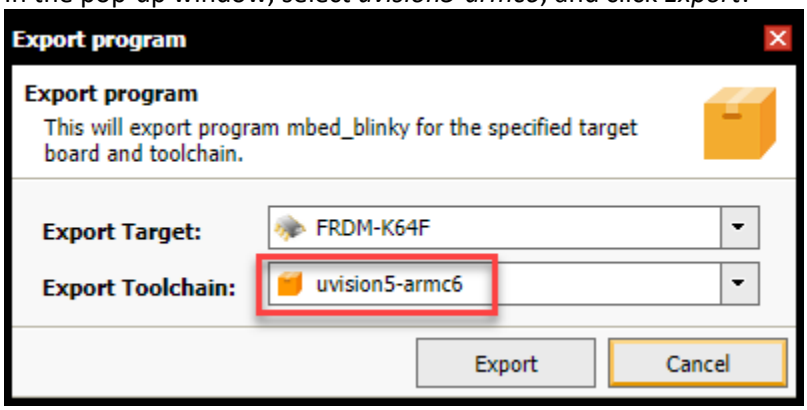
1. Open the Online Compiler and verify that you have selected the right development board (in the top-right corner).



2. Right-click on your project, select *Export Program...*



3. In the pop-up window, select *uvision5-armc6*, and click *Export*:



4. A ZIP file containing Mbed OS and the application is downloaded automatically.

Import the project in µVision

1. Unpack the ZIP file and switch to the directory containing the project.
2. Double-click on the uvprojx file to open it in µVision.
3. [Optional] If µVision prompts you about a missing device pack, download and install the required pack.

Build the project

1. Go to **Project – Build Target (F7)** to compile the project.

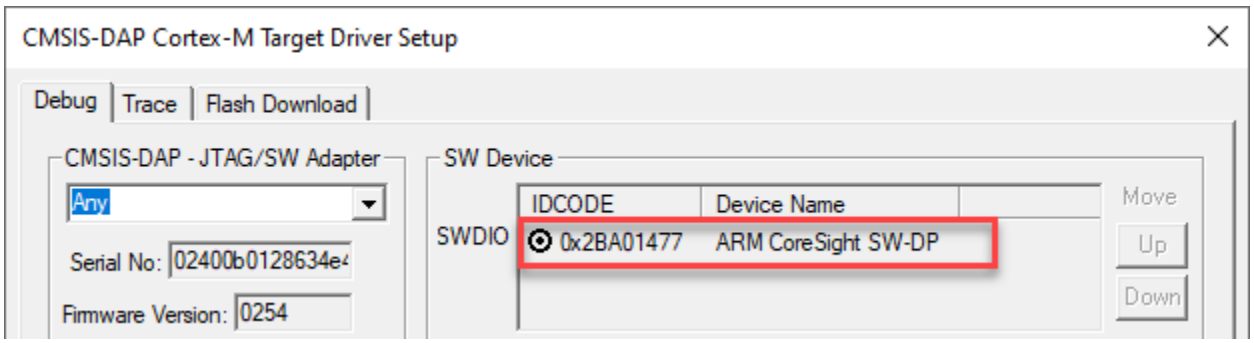
   Note

   This step compiles a lot more files than you might expect. Mbed OS ships with a very wide variety of components, and these are all built the first time you compile. This does not bloat your build though; unused components will be removed by the Arm linker.
2. The build should finish without errors (warnings may occur).

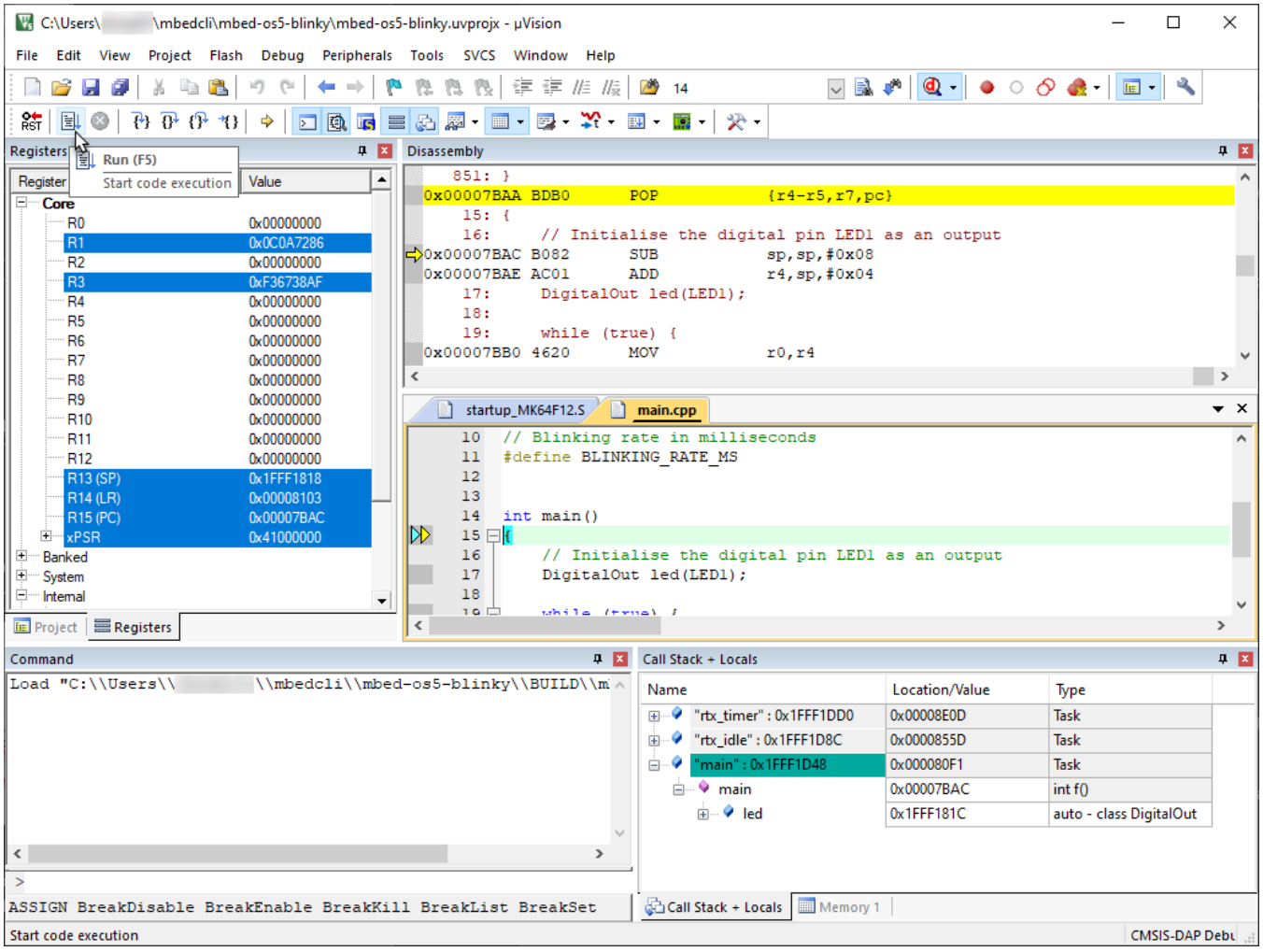
Download the project and start debugging

Once the build has finished, verify that the correct debug adapter is selected.

1. Go to **Project – Options for Target** and go to the **Debug** tab. Select the appropriate debug adapter that is available on your target hardware and click on **Settings**. Make sure that the **SW Device** area shows an **IDCODE**:



   Click OK twice to exit the dialog.
2. Go to **Debug – Start/Stop Debug Session (Ctrl+F5)** to download the application to the target. The µVision debug view opens and the program runs to main and stop there.
3. Use **Debug – Run (F5)/Step (F11)/Step Over(F10)** to control the program.



Running the imported Mbed OS 5 project in µVision

Notes and troubleshooting

The Mbed OS configuration system

Mbed OS uses an advanced configuration system which allows for inheritance and per-target configuration. This configuration is specified in `mbed_lib.json` and `mbed_app.json` files. Because µVision cannot parse .json files, another file with macro definitions – based on these files – is placed in `mbed_config.h`. We do not recommend changing this header file yourself, but rather to re-export the project when a change in configuration is made. This will make it much easier to update Mbed OS in the future.

More information on the Mbed OS configuration system can be found at https://os.mbed.com/docs/latest/reference/configuration.html.

Internal command error on ST-Link

For some development boards with an on-board ST-Link debug adapter, the error 'internal command error' is thrown when trying to load a program. This is because the wrong clock frequency is selected by default.

Go to **Project – Options for Target** – **Debug** tab to fix this. Next to **ST-Link Debugger** click **Settings**. Under **Target Com** switch to **JTAG**, then switch back to **SWD**. This will automatically select the right clock frequency: