

Abstract

This application note demonstrates how to build Mbed OS 5 projects with Keil μ Vision for development and debugging.

Contents

| | |
|---------------------------------------------------|----|
| Abstract | 1 |
| Introduction..... | 1 |
| Prerequisites..... | 1 |
| Importing an existing project | 2 |
| Importing via the Online Compiler | 2 |
| Importing via Mbed CLI..... | 4 |
| Updating the Mbed OS version..... | 4 |
| Building and flashing..... | 4 |
| Starting a new project with Mbed CLI..... | 6 |
| Building and flashing..... | 6 |
| Adding new libraries..... | 9 |
| Notes and troubleshooting..... | 11 |
| The Mbed OS Configuration system | 11 |
| Build profiles and optimizations | 11 |
| Internal Command Error on ST-Link interface | 11 |

Introduction

Arm Mbed OS is a free, open-source embedded operating system designed specifically for the "things" in the Internet of Things. 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. It is compatible with over 140 development boards and modules and has a vibrant developer community with thousands of libraries available.

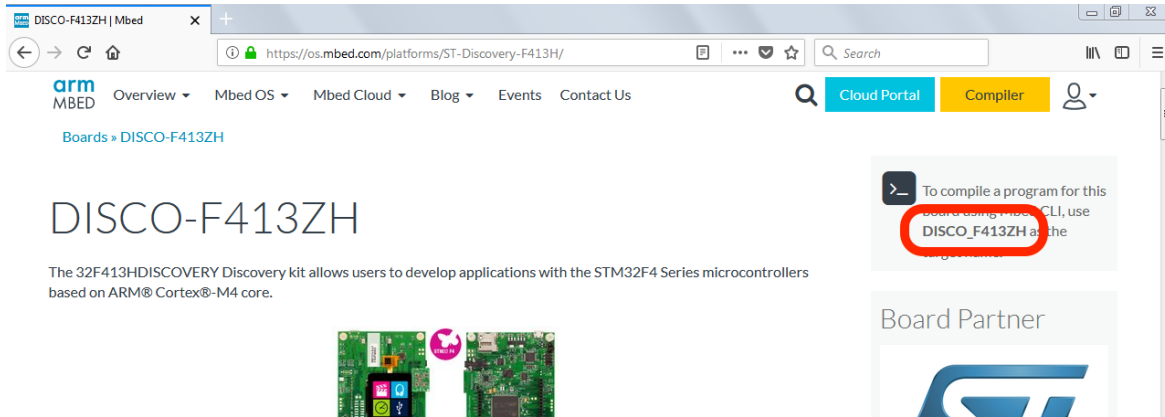
Mbed OS is not available as an MDK Middleware package, but has its own configuration and dependency management system. This application note covers how to get started building Mbed OS projects with μ Vision, how to add libraries or features, and how to migrate projects from the Online Compiler to your offline toolchain.

Prerequisites

- Keil MDK-ARM version 5.0 or later – the latest evaluation version can be found here:
www.keil.com/demo/eval/arm.htm.
- A valid MDK-Professional license – a free 7-day license can be obtained here:
www.keil.com/mdk5/activation/.

- An Mbed-enabled development board – a list is available here: <https://os.mbed.com/platforms/>.
- Mbed CLI, the build and dependency management tools for Mbed OS. Installation instructions are here: <https://os.mbed.com/docs/latest/tools/installation-and-setup.html>.

You will also need to know the target name of your development board. You can find this by finding your development board at <https://os.mbed.com/platforms> and clicking through to the detail page. The target name is listed in the upper right corner.



Finding the target name for your development board

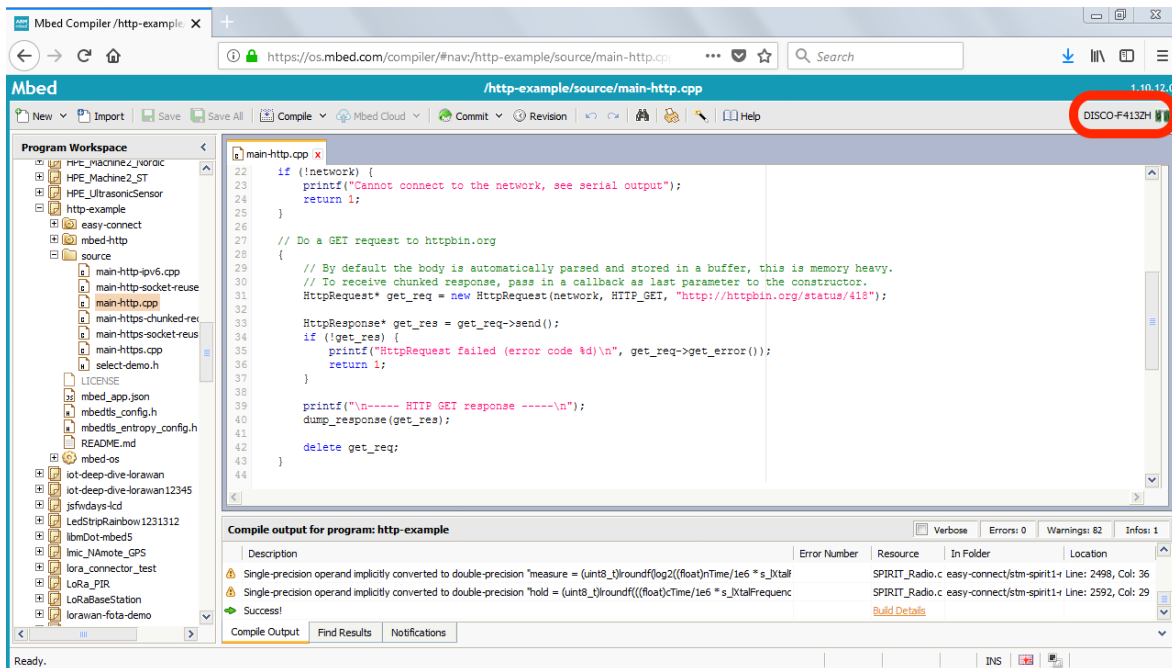
Places where you need to fill the target name in are marked with **YOUR_TARGET_NAME** in this application note.

Importing an existing project

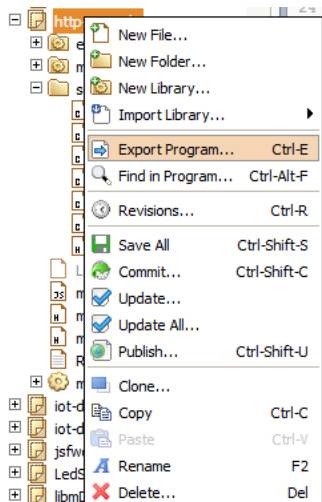
Importing via the Online Compiler

The Mbed Online Compiler – at <https://os.mbed.com/compiler> – is the way that most first-time users experience Mbed OS. It's 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.



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



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

4. A zip file containing Mbed OS and the application downloads. Unzip it.

Exported projects contain only a subset of Mbed OS 5 and cannot be managed by Mbed CLI. To resolve this:

5. Open a command prompt or terminal window, and navigate to the location where you unzipped the folder.

6. Run:

```
rmmdir mbed mbed-os /S
```

7. Then run:

```
mbed deploy
```

8. This adds the full source of Mbed OS to your project. Re-create the μ Vision project by running:

```
mbed export -i uvision5 -m YOUR_TARGET_NAME
```

9. **Optionally:** Continue to 'Updating the Mbed OS version' to make sure you have the latest version of Mbed OS.
10. Continue to 'Building and flashing'.

Importing via Mbed CLI

A project can also be imported directly through Mbed CLI. Open a command prompt or terminal window, and navigate to the folder where you want to create the project.

1. Find the project you want to import, and run:

```
mbed import https://os.mbed.com/teams/mbed-os-examples/code/mbed-os-example-blinky
```

2. This downloads the project and all dependencies.
3. In the command prompt navigate to the imported project, and create the MDK project by running:

```
mbed export -m YOUR_TARGET_NAME -i uvision5
```

Updating the Mbed OS version

Imported projects might contain an outdated Mbed OS version that does not contain the latest fixes or security updates. You can update the Mbed OS version through Mbed CLI:

1. Remove the current Mbed OS version via:

```
rmdir mbed mbed-os /s
```

2. Remove the current library files for Mbed OS via:

```
del mbed.bld mbed-os.lib mbed.lib
```

3. Update to the latest version of Mbed OS via:

```
mbed add mbed-os
```

4. Recreate the μ Vision project via:

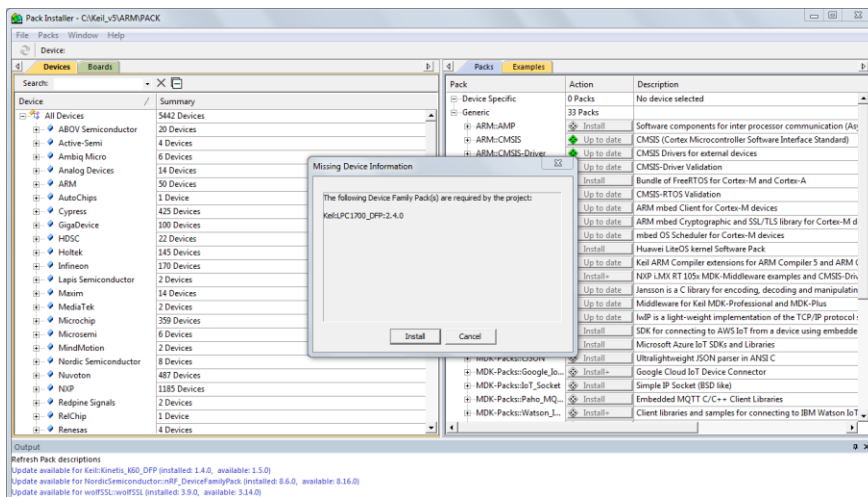
```
mbed export -m YOUR_TARGET_NAME -i uvision5
```

You're now using the latest version of Mbed OS.

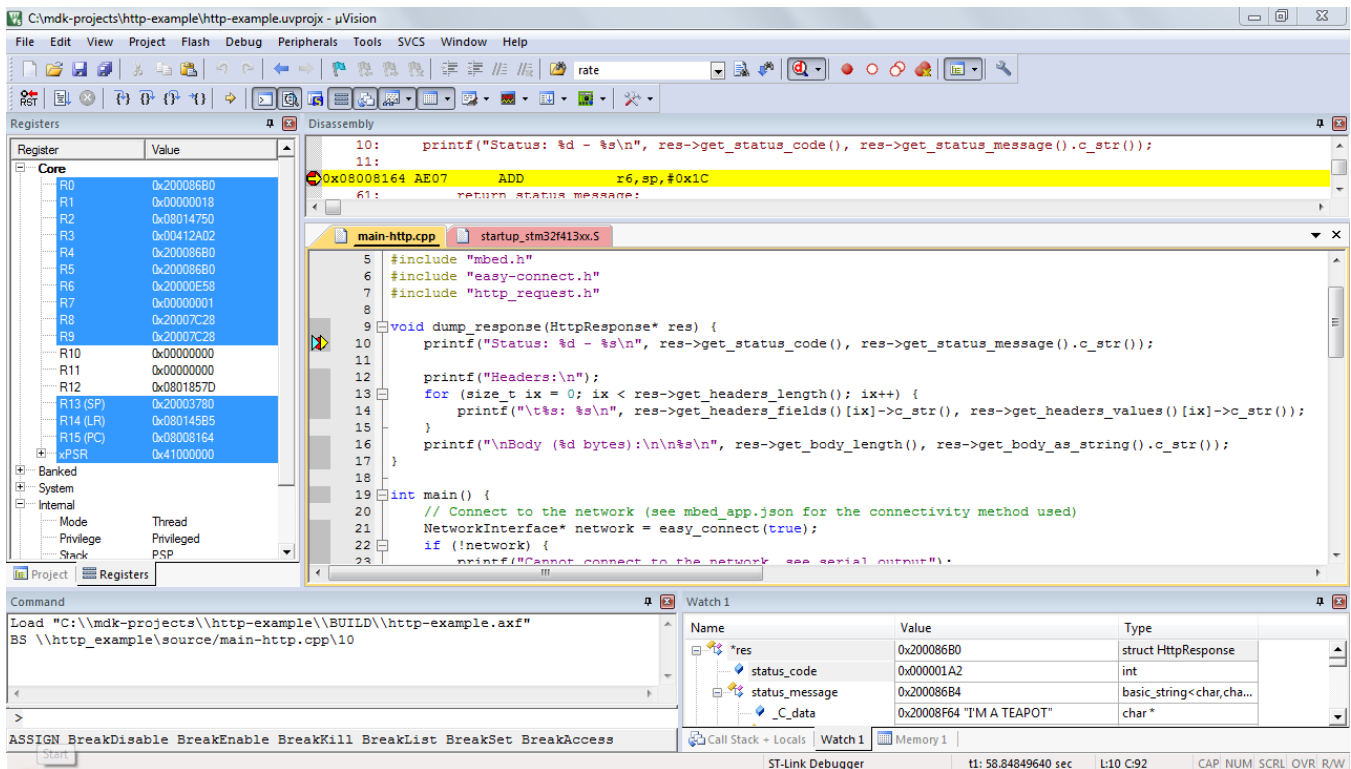
Building and flashing

After importing the project double-click on the uvprojx file to open μ Vision.

1. If μ Vision prompts you about a missing device pack, click *OK* to install it.



2. Build the project by clicking *Project > Build Target*.
 - a. Note that 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 linked out.
3. After building succeeds, click *Debug > Start/Stop Debugging session* to flash the target, and run the application.



Running the imported Mbed OS 5 project in µVision

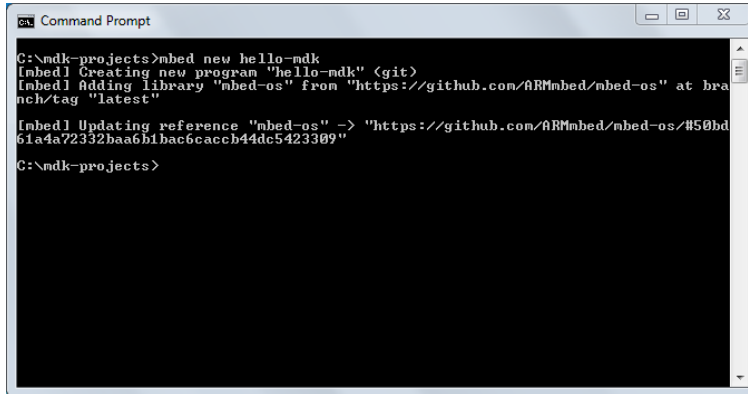
Starting a new project with Mbed CLI

To start a new Mbed OS project, open a command prompt or terminal window:

1. Navigate to the folder where you want to create the new project.
2. Run:

```
mbed new hello-mdk
```

3. This creates a new folder with the name 'hello-mdk' and adds the latest stable Mbed OS release.



```
Command Prompt
C:\mdk-projects>mbed new hello-mdk
[mbed] Creating new program "hello-mdk" <git>
[mbed] Adding library "mbed-os" from "https://github.com/ARMmbed/mbed-os" at branch/tag "latest"
[mbed] Updating reference "mbed-os" -> "https://github.com/ARMmbed/mbed-os/#50bd61a4a72332baa6b1bac6acac44dc5423309"
C:\mdk-projects>
```

4. Connect your development board to your computer.
5. In the terminal navigate to the 'hello-mdk' project, and create the MDK project by running:

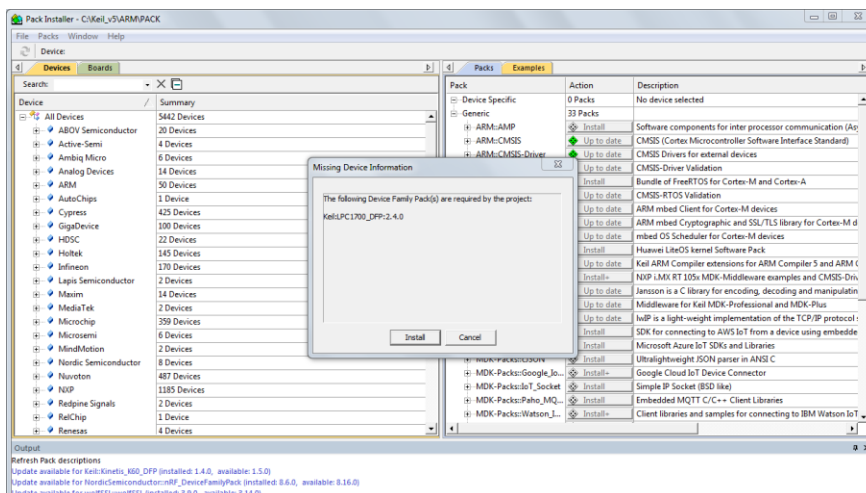
```
mbed export -m YOUR_TARGET_NAME -i uvision5
```

6. This will have generated the 'hello-mdk.uvprojx' file.

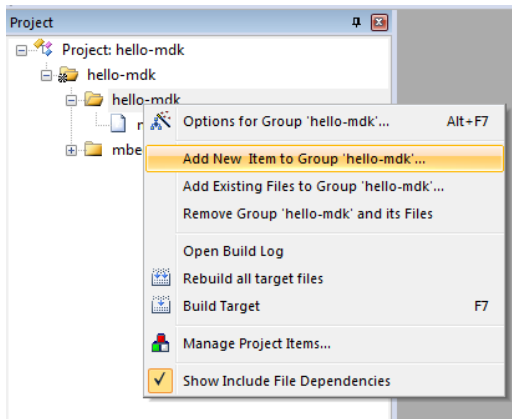
Building and flashing

After you imported or created a project you can build the project in µVision.

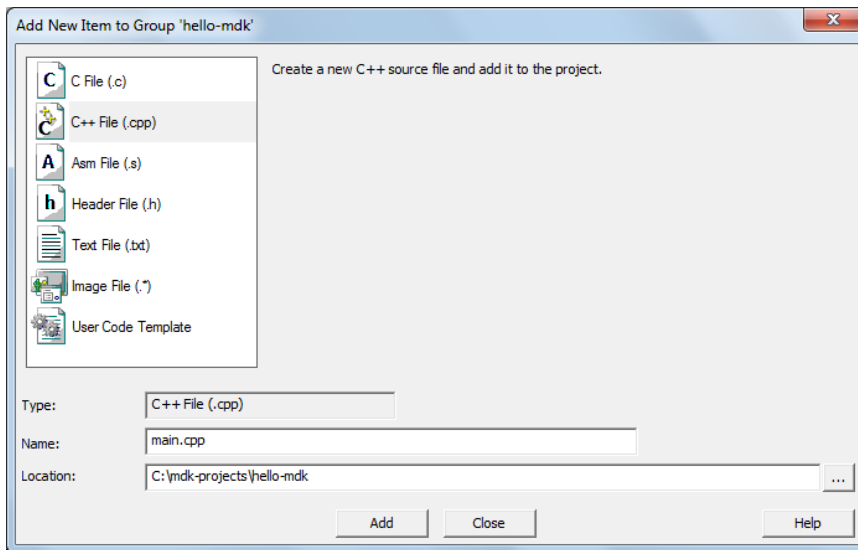
1. Open 'hello-mdk.uvprojx' to open µVision.
2. If µVision prompts you about a missing device pack, click OK to install it.



3. Right-click on 'hello-mdk' in the 'Project' pane and click *Add New Item to Group 'hello-mdk'*.



4. Choose 'C++ File', name it 'main.cpp' and click *Add*.



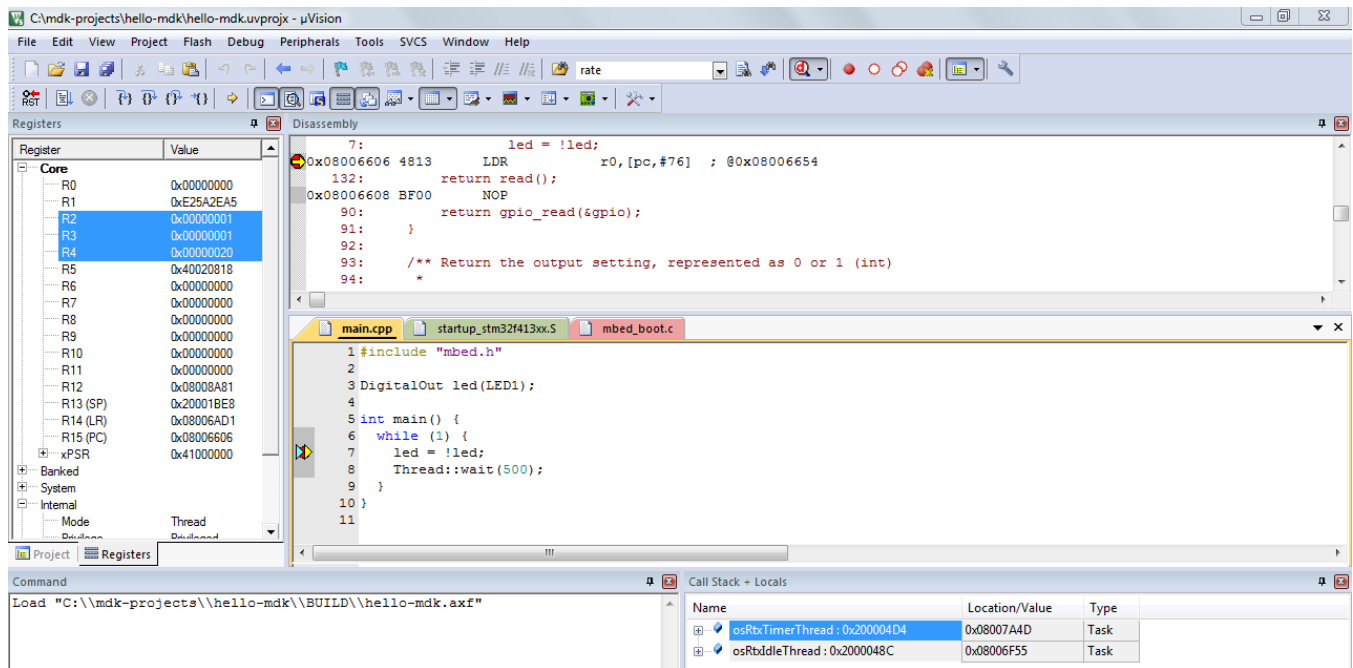
5. Add the following code to the new file:

```
#include "mbed.h"

DigitalOut led(LED1);

int main() {
    while (1) {
        led = !led;
        Thread::wait(500);
    }
}
```

6. Build the project by clicking *Project > Build Target*.
 - a. Note that 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 linked out.
7. After building succeeds, click *Debug > Start/Stop Debugging session* to flash the target, and run the application.

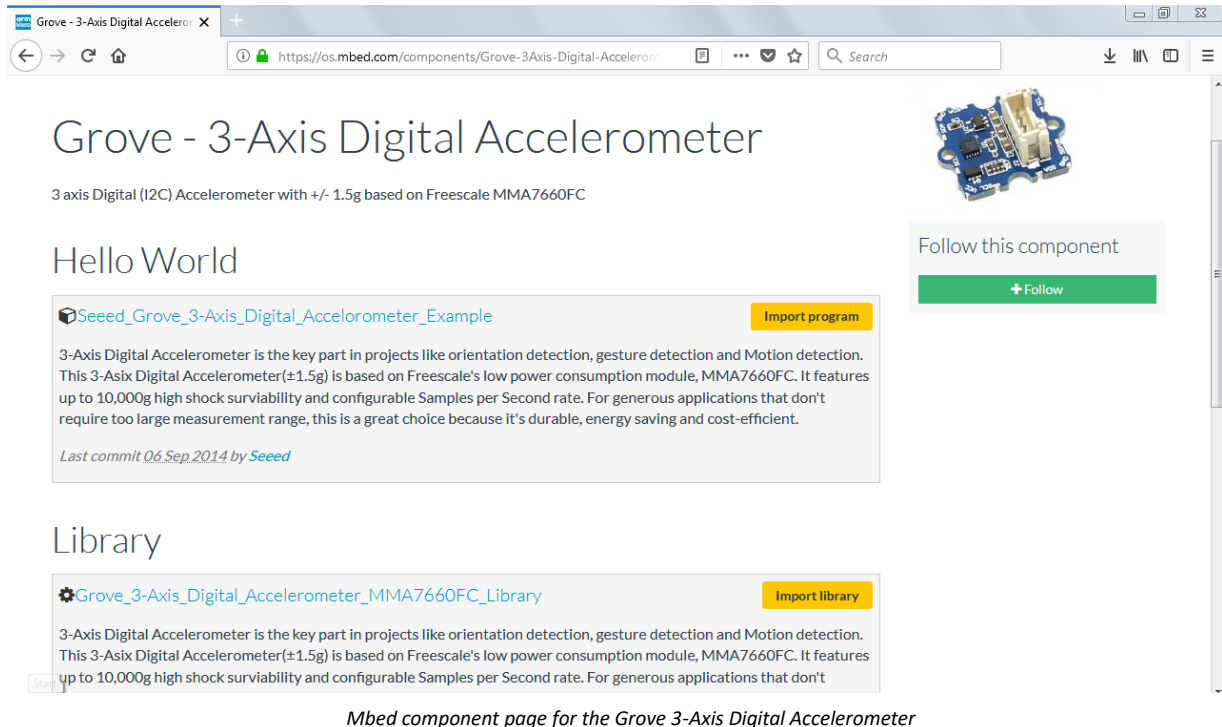


Debugging a new Mbed OS 5 project in μ Vision

Adding new libraries

The Mbed developer website - at <https://os.mbed.com/> - contains thousands of libraries for a wide set of components, connectivity methods and algorithms. These libraries can be added and maintained through Mbed CLI. This section shows how to add a new sensor library to your μ Vision project. Here the Grove 3-Axis Digital Accelerometer is used, but the same principle applies to many other sensors.

1. Find the sensor on the Mbed developer website. It's located at <https://os.mbed.com/components/Grove-3Axis-Digital-Accelerometer/>.



Grove - 3-Axis Digital Accelerometer

3 axis Digital (I2C) Accelerometer with +/- 1.5g based on Freescale MMA7660FC

Hello World

[Seed_Grove_3-Axis_Digital_Accelerometer_Example](#) [Import program](#)

3-Axis Digital Accelerometer is the key part in projects like orientation detection, gesture detection and Motion detection. This 3-Axis Digital Accelerometer ($\pm 1.5g$) is based on Freescale's low power consumption module, MMA7660FC. It features up to 10,000g high shock survivability and configurable Samples per Second rate. For generous applications that don't require too large measurement range, this is a great choice because it's durable, energy saving and cost-efficient.

Last commit 06 Sep 2014 by Seed

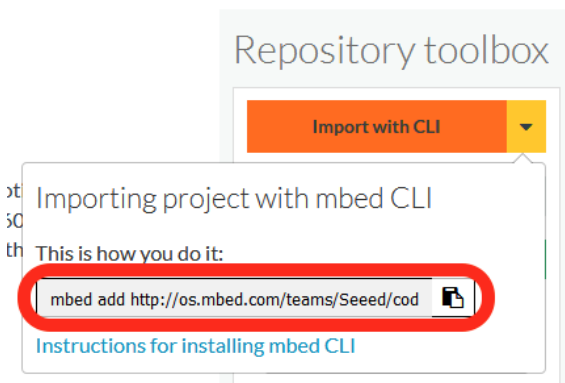
Library

[Grove_3-Axis_Digital_Accelerometer_MMA7660FC_Library](#) [Import library](#)

3-Axis Digital Accelerometer is the key part in projects like orientation detection, gesture detection and Motion detection. This 3-Axis Digital Accelerometer ($\pm 1.5g$) is based on Freescale's low power consumption module, MMA7660FC. It features up to 10,000g high shock survivability and configurable Samples per Second rate. For generous applications that don't

Mbed component page for the Grove 3-Axis Digital Accelerometer

2. Both a library and an example program are listed on this page. Click through to the library at https://os.mbed.com/teams/Seed/code/Grove_3-Axis_Digital_Accelerometer_MMA76/.
3. Under 'Repository toolbox', select 'Import with CLI' and copy the command.



Repository toolbox

[Import with CLI](#)

Importing project with mbed CLI

This is how you do it:

```
mbed add http://os.mbed.com/teams/Seed/cod
```

[Instructions for installing mbed CLI](#)

4. In the command prompt or terminal, navigate to your project folder and run the command:

```
mbed add http://os.mbed.com/teams/Seed/code/Grove_3-Axis_Digital_Accelerometer_MMA76/
```
5. This fetches the latest version of the library.

6. To add this project to μ Vision, re-run the export command:

```
mbd export -m YOUR_TARGET_NAME -i uvision5
```

Note: This re-creates the μ Vision project file, and settings might be overwritten. Alternatively, you can manually add the files in the 'Grove_3-Axis-Digital-Accelerometer-MMA76' to your project.

7. You can now reference the library from code; and build and debug as usual.

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 this file, a 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>.

Build profiles and optimizations

Mbed OS uses build profiles to enable compiler features, to omit / include debug information, and to determine the optimization level. When creating the uVision project the 'debug' profile is used by default. If you want to optimize for size, enable C++11 support or use newlib-nano you can override this when creating the uVision project. You do this via:

```
mbed export -m YOUR_TARGET_NAME -i uvision5 --profile profile-name
```

Mbed OS ships with 'debug', 'develop' and 'release' profiles, but custom profiles can also be created. More information on build profiles can be found at <https://os.mbed.com/docs/latest/tools/build-profiles.html>.

Some other notes on optimizing the RAM and ROM footprint of Mbed OS applications, and notes about stack and heap tracing, are at <https://os.mbed.com/blog/entry/Reducing-memory-usage-with-a-custom-prin/>.

Internal Command Error on ST-Link interface

For some development boards with an ST-Link debugging interface, the error 'internal command error' is thrown when trying to flash. This is because the wrong clock frequency is selected by default. You can fix this by going into *Project > Options for Target > Debug*. Click *Settings*, next to ST-Link. Under 'Target COM' switch to 'JTAG', then switch back to 'SWD'. This will automatically select the right clock frequency.

