MCUXpresso IDE Installation Guide

Rev. 10.2.0 — 14 May, 2018 User guide

14 May, 2018

1. Installation

MCUXpresso IDE will install with a base set of drivers and built-in support for a large range of LPC MCUs and native debug connections via LinkServer (CMSIS-DAP). Additional part (MCU) support can be added at any time by downloading and installing the required SDK packages.

Support for SEGGER J-Link Debug probes and P&E Micro Debug probes is also installed by default.

Note: As of MCUXpresso IDE version 10.2.0, only one product variant is available, replacing the previous Free and Pro Editions. MCUXpresso IDE now includes *out-of-the-box* all features previously restricted to the Pro Edition. It does not require any activation procedure and contains no limitations on build or debug code sizes.

1.1 Host Computer Requirements

Before installing MCUXpresso IDE, you should make sure your development host computer meets the following requirements:

- A standard x86 host with 4GB RAM minimum, 4GB of available disk space, a recommended screen resolution of at least 1440x900 and running one of the operating systems specified below.

An internet connection is required for the downloading of SDKs, product updates and for the use the Config Tools.

1.2 Windows

- Microsoft® Windows 7, 8.1, 10
  - Both 32-bit and 64-bit Windows are supported.

MCUXpresso IDE is installed into a single directory, of your choice. Unlike many software packages, MCUXpresso IDE does not install or use any keys in the Windows Registry, or use or modify any environment variables (including PATH), resulting in a very clean installation that does not interfere with anything else on your PC. However, third party debug probe support code may make such modifications.

Should you wish to use the command-line tools, a command file *MCUXpressoPath.cmd* is provided to set up the path for the local command window. This file is located within:

```
<install_dir>\ide\bin\scripts
```

Note: During the installation, you will be prompted to install a variety of drivers. These are required for correct operation, and include:

- Philips (NXP) Universal Serial Bus
  - This can take some time to complete
- Jungo Connectivity and Jungo Ltd
  - These are installed by the P&E debug plugin
- Freescale P&E Micro
- Ashling/NXP

The installer will also silently install drivers for:

- SEGGER
- LPC-Link
- RedProbe+

- RDB-Link

After the installation has completed, in order to use some Kinetis boards with OpenSDA mbed CMSIS-DAP debug connection and LPCXpresso Max boards, an mbed Serial Port driver is required. This can be downloaded via the IDE link at:

```
Help -> Additional Resources -> MBED Serial Port Driver Website
```

Without this driver, the mbed based debug probe will not be found.

1.3 Mac OS X

- Mac OS X 10.11 (or later)
  - MCUXpresso IDE **may** work on older versions but we cannot provide support if it does not.

The MCUXpresso IDE installer is supplied as a Mac OS X *.pkg* installer file. Double-click on the installer to install MCUXpresso IDE into a subfolder of your Applications folder.

To start MCUXpresso IDE, use the Mac OS X Launchpad. Alternatively, click the **Open MCUXpresso IDE** icon in the */Applications/MCUXpresso IDE_version* folder or run **MCUXpresso IDE.app**, which can be found in the *MCUXpresso IDE* subfolder of the main MCUXpresso IDE installation directory within */Applications*.

1.3.1 Mac OS X Sierra – Unexpected Delays during Debug Operations

Mac OS X Sierra installations may experience an approximately 5-second delay when a call to get the localhost address is made from Java. This call is used within MCUXpresso IDE when establishing a debug connection for LinkServer, P&E or SEGGER. Therefore, if a delay is experienced during a debug operation, the following fix may be required.

First, lauch a Terminal session and enter the command *hostname*, which will return something of the form

```
~$ hostname
user-Mac.local
```

Next, edit your */etc/hosts* file and add the returned hostname after the localhost entries as shown below.

```
127.0.0.1   localhost user-Mac.local
::1         localhost user-Mac.local
```

Save the *etc/hosts* file. Once this change has been made, the 5-second delay should be reduced to a few milliseconds.

Further information about this issue can be obtained by performing a web search of the form *MacOS Sierra Java localhost slow*.

1.4 Linux

- Linux – Ubuntu 16.04 LTS and later
  - Only 64-bit versions of Linux are supported.

MCUXpresso IDE for Linux is a 64-bit application, so it will not run on 32-bit systems. It is supported and tested only on the Linux distribution Ubuntu 16.04 LTS. Limited testing has also been done on pre-release builds of Ubuntu 18.04 LTS.

The installer is supplied as an executable that installs the MCUXpresso IDE components. The installer requires root privileges, although, once it is installed, no special privileges are required to run the MCUXpresso IDE. The installer will request a super-user password when it is started. Once installation has completed, we strongly recommend that your system is restarted – if you do not do this, then some areas of the tools may not function correctly.

For further details, please see Appendix A – Linux Installation [8]

1.4.1 Other Linux Distros

Due to the huge variation in capabilities of different Linux distributions and versions, MCUXpresso IDE **may** work on other distributions / versions but we cannot provide support if it does not.

In such circumstances, the MCUXpresso IDE forum is a good place to search for information or to post questions, as other users may be able to assist you.

1.5 Installation Notes

1.5.1 High-Resolution Displays

When using high-resolution displays, high-dpi icons can be selected by adding an extra argument *-Dswt.autoScale=200* to the end of the mcuxpresso.ini file.

This can be found, for example, at:

```
Windows: C:\nxp\<install_dir>\ide\mcuxpressoide.ini
Mac: /Applications/<install_dir>/ide/MCUXpressoIDE.app/Contents/MacOS/mcuxpressoide.ini
```

Note: This may cause the startup splash screen to be cropped, but it will not affect the product's usability.

1.5.2 Running under Virtual Machines

It is possible to install the MCUXpresso IDE within a virtual machine (VM) environment. Generally such installations cause few issues. Due to the nature of VMs, the most likely problems relate to sharing of resources (USB, memory).

In the unlikely event that you experience issues, we welcome reports, but due to the nature of VM operation we can offer no guarantee of resolution.

1.5.3 Help us improve MCUXpresso IDE

MCUXpresso IDE can send anonymous information to NXP on how you use the IDE, including the built-in Config Tools, and with which MCUs. This information can help us to improve the functionality of the tools as well as to resolve problems. You can turn this information collection off at any time by unticking the workspace option:

Preferences -> MCUXpresso IDE -> General -> Help us improve the tool

2. Activation

There is no activation process required for the use of MCUXpresso IDE, all features are available after installation.

3. Migrating from an earlier version of MCUXpresso IDE

MCUXpresso IDE functions and features are under continuous development, it is strongly recommended for user to read both the ReadMe and KnownIssues files inside the product installation directory.

We would generally recommend the following flow when a new IDE release becomes available....

Install the new release in parallel with the original version. This allows you to evaluate the new release before committing to using it for your main product development work.

When using the new version, use a new workspace – to keep your new “evaluation” world separate from your old “development” world. You can very easily copy your projects from your existing workspace to your new workspace – for example by checking them out of version control again, or simply using the IDE’s Quickstart Panel option to “Import project(s) from file system...” and pointing at the root directory of your existing workspace.

Note: MCUXpresso IDE v10.2 projects are not backward compatible with earlier versions of MCUXpresso IDE. Also, if an existing project is edited with MCUXpresso IDE v10.2, it may no longer be useable with an earlier version of MCUXpresso IDE.

Preferences can also be imported from your existing workspace to your new workspace. To do this run your old IDE installation, and use **File -> Export -> General -> Preferences** to export. Then in your new IDE installation use **File -> Import -> General -> Preferences** to pull your preferences in.

If you have installed additional plugins into your original IDE installation, then you can also import these. To do this, from your new installation select **File -> Import -> Install -> From Existing Installation** and point at the *ide* directory within your original IDE’s installation directory. **Note**: that on Mac OS X / Linux – this option will effectively run automatically the first time you run the new version of the IDE.

Tip

MCUXpresso IDE version 10.2.0 allows projects to be imported by simply dragging a project folder (or zip archive of projects) directly into the IDE’s Project Explorer view. In addition, it is possible to drag from the Project Explorer view of an older IDE, directly into Project Explorer view of MCUXpresso IDE version 10.2.0. This provides a very simple way of transfering projects into the new IDE. However, it is recommended that build configuration and launch configuration folders are deleted before (or after) copying. **Note**: Due to enhancements in MCUXpresso IDE version 10.2, older launch configurations are no longer compatible with this version, failure to delete them will lead to a warning and the launch configuration will then be deleted automatically on the next debug attempt.

4. Differences from LPCXpresso IDE

MCUXpresso IDE functions and features are under continuous development, it is strongly recommended for user to read both the ReadMe and KnownIssues files installed with the product.

MCUXpresso IDE is built on upon technology contained within the LPCXpresso IDE. This means that for users familiar with LPCXpresso IDE, the new MCUXpresso IDE will look relatively familiar. However, there are some differences to be aware of. Some of these are detailed below.

Note: MCUXpresso IDE projects are not backward compatible with LPCXpresso IDE. Also, if an LPCXpresso IDE project is edited with MCUXpresso IDE, it may no longer be useable within LPCXpresso IDE.

As of MCUXpresso IDE version 10.2.0, only one product is available replacing both the Free and Pro editions.

MCUXpresso IDE now includes all features previously restricted to the Pro Edition, it does not require any activation procedure and contains no limitations on build or debug code sizes.

4.1 New Features Include...

4.1.1 Part Support

Support for MCUXpresso SDKs allows runtime addition of part knowledge and example/driver code. This allows use of MCUXpresso IDE with Kinetis MCUs and new LPC MCUs, as well as existing preinstalled LPC MCUs.

4.1.2 Debug Capabilities

There is built-in support for debugging via P&E Micro and SEGGER J-Link debug probes, in addition to the native LinkServer debug technology inherited from LPCXpresso IDE.

LPC-Link2 performance has been enhanced.

There is a new Global Variables view, with Live Variables functionality with LinkServer debug connections.

FreeRTOS debug features have been added.

4.1.3 Config Tools

The IDE now installs with additional fully integrated configuration tools comprising Pins, Clocks and Peripherals Tools which are exclusively designed to work with SDK projects.

4.1.4 General Product

MCUXpresso IDE is based on updated versions of Eclipse/CDT and GCC tools. It is supplied with the eGIT plugin built in.

4.1.5 Application Memory Usage

A new optional “MCUXpresso Style” of heap and stack placement is supported, allowing the heap and stack to be included in the overall application memory usage information generated by the link step of the build. An Editor also allows heap and stack to be easily relocated to different memory regions.

Redlib no longer uses the heap for part of the IO stream buffers, meaning that the full steam buffer usage is represented in the memory usage information generated by the link step of the

build. This can appear to increase the BSS size of an image, but overall it means that less RAM is used. This also removes the need for *malloc()* routines in some applications, reducing code size.

A new Redlib “semihost-nf” (no files) variant reduces the number of IO stream buffers to just stdin/stdout/stderr – allowing a noticeable reduction in the BSS size of the image for applications that do not need to use files on the host PC over the semihosting mechanism.

4.1.6 Activation and Licensing

MCUXpresso IDE v10.2 makes no use of license activation. All features are enabled out-of-the-box.

4.2 Removed Features Include...

- No support for LPC2000 and LPC3000 (ARM7/ARM9) families.
- Legacy SPIFI flash drivers for LPC18_43
  - Use LPC18_43_SPIFI_GENERIC.cfx or LPC18_43_SPIFI_JEDEC.cfx instead.
- No support for Red State (the state machine editor and code generator).

5. Appendix A – Linux Installation

5.1 Ubuntu

The product is distributed as a file called `mcuxpressoide-<build>.x86_64.deb.bin`, which is a binary file that when run will create a Debian package and install it.

To install this file, it must be made executable and then run as root. For example, if the file is in the current working directory:

```
chmod +x mcuxpressoide-<build>.x86_64.deb.bin
sudo mcuxpressoide-<build>.x86_64.deb.bin
```

Once you have agreed to the license terms the Debian package will be installed along with any packages that it requires.

5.1.1 Creating a Backup

To create a backup of an older version during installation, use `-b` or `--backup`. This needs to be passed to the underlying script of the `.run` package by calling:

```
<install_package>.deb.bin -- -b or
<install_package>.deb.bin -- --backup
```

5.2 Other Linux Distributions

Other distributions are not supported or tested. The Debian package lists other package names as dependencies, and these may not be among the packages provided by all distributions. Nonetheless, it may run on other Linux distributions.

- For distributions based on Debian (with a Debian-based package manager), try the `.x86_64.deb.bin` installation image.

5.3 Running the MCUXpresso IDE

5.3.1 From the Desktop

To run from the desktop, search for a program containing “MCUXpresso” in its name and run it as normal for your desktop. It is usually found in the “Development” application category. (This should work in most Linux Desktop environments.)

Note: The installation script now creates a softlink at `/usr/local/mcuxpressoide` pointing to the real installation directory.

5.3.2 From bash

The product is installed in the directory `/usr/local/mcuxpressoide-<build>` and can be run using the command `mcuxpressoide` if `/usr/local/mcuxpresso-<build>/ide` is placed on your path, for example using:

```
export PATH=”/usr/local/mcuxpresso-<build>/ide:$PATH”
```

Depending on the desktop manager you use you may need to set some environment variables. It is safe to use these settings for any desktop, however, and you can always run using the following command line.

```
SWT_GTK3=0 UBUNTU_MENUPROXY=0 mcuxpressoide &
```

5.3.3 Further Information

- `SWT_GTK3` controls the use of your distribution's GTK libraries that are used in Gnome-based desktops (including Ubuntu Unity and Gnome Desktop). The setting above stops GTK3 from being used in the IDE. The version of Eclipse underlying MCUXpresso IDE will show small errors that will make normal working impossible in these desktops unless this setting is used.
- `UBUNTU_MENUPROXY` controls the way in which the menu bar of an application can appear at the top of the screen even when an application is not used in full-screen mode. Some users have reported issues in Eclipse in some window managers when the setting above is not used (although we have not observed them ourselves).

5.3.4 Known Issues

On (at least) Ubuntu 16.10 the names of the boards underneath their photos do not appear.

6. Appendix B – Migrating from LPCXpresso IDE version 8.2.x – Hints and Tips

6.1 Introduction

MCUXpresso IDE incorporates core technology from LPCXpresso IDE 8.2.2.

Migrating code from LPCXpresso IDE to MCUXpresso IDE should be straightforward, though you should always browse the release notes, the supplied documentation and the online FAQ material.

Below are some hints and suggestions of things that you should do or consider when migrating.

6.1.1 Parallel Installations

A new version of the MCUXpresso IDE may be installed in parallel with existing installations and also in parallel with LPCXpresso IDE. This allows a newly released version to be tried alongside a version currently installed.

Furthermore, there is no need to take any special care with licenses (activation codes), since any installed code will automatically be picked up by the new MCUXpresso IDE installation.

6.1.2 Installing Eclipse Plugins

If you install a new version of MCUXpresso IDE on Mac OS X or Linux, then the first time you run the new product you will be offered the opportunity to reinstall previously used plugins (for example, those for version control). However, this does not happen on Windows, and manually installing your favorite plugins may take considerable time to complete.

An alternative approach is to import the plugins from an earlier LPCXpresso IDE installation. To do this, follow:

```
File->Import->Install->From Existing Installation
```

Then browse to the `lpcxpresso` directory within an existing LPCXpresso IDE application's installation.

6.1.3 Managing Workspaces

Whilst a new MCUXpresso IDE version can open workspaces created by an earlier release, a workspace (and the projects it contains) that have been used by a new MCUXpresso IDE version may not correctly load into an earlier version. Thus we would strongly recommend that you back up your projects before commencing any migration.

The simplest way to do this is to create a new workspace in the new MCUXpresso IDE version, and then import any projects into this new workspace. How to import projects into a new workspace is detailed in the FAQ

https://community.nxp.com/message/630625

Alternatively, if you have your projects checked into a version control system (for example, using Subversion and the Subclipe Eclipse plugin), then you can simply check your projects out into the new workspace.

You should also ensure that you do a full, clean build after switching to the new version.

6.1.4 Launch Configurations

Sometimes the contents of, or options specified in, the debug launch configurations used by MCUXpresso IDE can change between versions. Thus, when moving to a new version of the MCUXpresso IDE, we would recommend deleting any debug launch configurations within your project that were created by an earlier version. These files are typically named

```
<projectname> Debug.launch and <projectname> Release.launch.
```

The easiest way to do this is to right-click on the project in Project Explorer and select **Launch Configurations -> Delete Launch Configurations**. The IDE will then automatically create a fresh set of launch configurations the next time you start a debug session. Note that you may need to reapply any modifications you made to your launch configurations in your previous version of MCUXpresso IDE.

For more information on launch configurations, please see the FAQ Launch Configuration Menu at

https://community.nxp.com/message/630714

6.1.5 Startup Code

The startup code generated by MCUXpresso IDE can sometimes be updated between releases, often to support new tool features. We would thus strongly recommend that you consider updating your startup code to match the latest generated by the project wizard for the part that you are using.

6.1.6 Linker Scripting

In LPCXpresso IDE V7.9.0 and later, the linker script template mechanism was overhauled to provide a much more flexible and powerful means for the user to change the content of the linker script generated by the managed linker script mechanism.

If you are moving a project that uses a modified linker script from a version of LPCXpresso IDE prior to version 7.9.0, then please read the detailed FAQ on Freemarker Linker Script Templates at

https://community.nxp.com/message/630611

6.1.7 Compiler Symbols

LPCXpresso IDE projects generally would define the compiler symbol __**CODE_RED**. This could then be used in source code to determine if the LPCXpresso IDE was being used to build the code, and conditionally compile sections of code in (or out) of the image being built.

When building under MCUXpresso IDE, the __**CODE_RED** symbol will not be removed from existing LPCXpresso IDE generated projects (for instance LPCOpen examples), either already in your workspace or in projects that you import into a new workspace. Also, if you create new projects for the preinstalled (LPC) MCUs, then, again, the symbol will be set up by the preinstalled MCU's new project wizards.

However, if you create projects for SDK installed MCUs, then the symbol __**CODE_RED** will not be set up for the compiler; the symbol __**MCUXPRESSO** is defined instead.

Thus, if you are porting existing code from LPCXpresso IDE into a new project created for an SDK installed MCU, then you need to check whether it is appropriate to change any instances of __**CODE_RED** to __**MCUXPRESSO**.

6.1.8 SPIFI Flash Drivers for LPC18xx and LPC43xx

Legacy SPIFI flash drivers, for example the `LPC18_43_SPIFI_1MB_64KB.cfx` or `LPC18_43_S25FL032P.cfx`, etc. have been removed from MCUXpresso IDE. In the last few releases of LPCXpresso IDE these drivers were in fact just copies of the `LPC18_43_SPIFI_GENERIC.cfx` driver, and were included to maintain compatability with certain older pre-built examples.

If you import a project for the LPC18xx or LPC43xx and experience an error because the SPIFI flash driver is not present, simply edit the project memory configuration and replace the missing driver with the `LPC18_43_SPIFI_GENERIC.cfx` driver.

6.1.9 License Compatibility with LPCXpresso IDE

MCUXpresso IDE requires no activation procedure and uses no licenses. Free or Pro Edition license from an LPCXpresso IDE install will have no impact on an MCUXpresso IDE installation.