

# **mbed FAQs**

## **Basic Questions**

### **Q. What is mbed?**

mbed is the industry's first online platform for fast, low-risk rapid prototyping of microcontroller-based systems.

### **Q. Why did you create mbed?**

The market for 32-bit microcontrollers is set for explosive growth, and the ability for new markets to adopt modern microcontroller technology will be a critical success factor for them. ARM has developed mbed to help these industries exploit the opportunities they present.

### **Q. How does mbed help me get started?**

The mbed tools enable new users who can get started with mbed tools in 60 seconds, by plugging in an mbed microcontroller, going to the [mbed.org](http://mbed.org) website to signup, and downloading and running a "Hello World!" binary just like saving to a USB Flash Drive. Compiling a first program takes only 60 seconds more: Launch the browser-based compiler, create a new template project, and click 'compile' to build and download the binary. The tools are in 'the cloud', there is nothing to configure or install, and everything works on Windows, Mac or Linux. For developers currently using proprietary 8/16-bit microcontrollers, discrete logic, or even those new to the industry, mbed removes the barriers to the world of modern 32-bit microcontrollers.

### **Q. How does mbed help me build prototypes?**

The mbed Microcontroller packages an NXP LPC1768 Cortex-M3 processor-based MCU and support components in a practical 40-pin 0.1" pitch DIP form-factor, ideal for experimenting on solderless breadboard, stripboard and through-hole PCBs. The mbed C/C++ Libraries provide high-level interfaces to microcontroller peripherals, enabling a clean, compact, API-driven approach to coding. The combination gives immediate connectivity to peripherals and modules for prototyping and iteration of microcontroller-based system designs, providing developers with the freedom to be more innovative and more productive.

### **Q. Who is mbed for?**

mbed has a broad appeal. Engineers new to embedded applications can use mbed to experiment and test product ideas that could benefit from advanced microcontrollers. Experienced engineers can be more productive and adventurous in the proof-of-concept stages of development. Marketing, Sales, Application Engineers and Distributors can use mbed as a consistent platform for demonstration, evaluation and support of microcontrollers, middleware and associated components. Education, Enthusiasts and Inventors can use mbed as an accessible way to experiment with the application of microcontrollers without worrying about implementation details. As a result, the mbed tools will help a diverse audience exploit the opportunities presented by advanced microcontrollers when introduced to their own area of expertise.

### **Q. How do I get one?**

The [mbed.org](http://mbed.org) website is now live, and the mbed Microcontroller based on the NXP LPC1768 is available for purchase. For purchasing details and to find out more about mbed, visit <http://mbed.org>

## **Specific Questions**

### **Q. All development boards provide code examples. How is mbed different?**

Whilst mbed does provide examples, mbed focuses on APIs and libraries; reusable functionality, with clear interfaces and solid implementations. Development board examples tend to prove something basically functions, whereas libraries give you the tools to get what you want done. The examples are not really a feature of mbed, more a consequence of the fact it is easy to create them.

### **Q. Can I use the code I develop with mbed for production? How much does it cost?**

The libraries are designed for prototyping as part of mbed; they are provided "as is," but could theoretically be used in a production system at no cost.

### **Q. Has it got a JTAG debugger?**

No, the mbed tools don't support JTAG debugging. Where this feature is deemed essential, mbed is not the solution. There are many great tools available from ARM partners within the Cortex Microcontroller Ecosystem from which to choose. With mbed, we've taken the approach of doing as much as possible to avoid firing up a debugger. The API-driven approach means our peripherals are all abstracted to their core concepts, and the implementations are reliable. With mbed, things like printf work out-of-the-box over the same USB interface, so this provides a easily accessible way to debug logical problems in a familiar fashion. You can even access the mbed USB filesystem to dump logfiles. Whilst sometimes it'll be sub-optimal, for the majority of the time, we've found it to be a good trade-off.

### **Q. Is there an offline compiler?**

No, everything is online. If you need to work offline, mbed is not the solution. Online tools avoid the installation, configuration or platform issues, and also has the advantage everyone is using the same setup, so official and community support works better. It won't be ideal all the time, but it works really well for a lot of people.

### **Q. Won't the free online compiler yield bloated, inefficient code?**

The mbed Compiler uses the same compiler technology found in the ARM RealView and Keil MDK tool suites, so it is the best in the industry. However, we are definitely taking advantage of the processing power and memory size of Cortex MCUs to enable us to throw software at the problem.

The mbed Libraries are focused on providing the abstractions needed to optimize time to prototype, rather than code size. But even still, combined with an optimizing compiler it really isn't much of an overhead. You can always go in and optimize the critical components, so mbed gives a great way to avoid falling in to the early optimization trap and only spend the effort where needed.

### **Q. Am I able to use my existing ARM application code with the mbed tool?**

The mbed Compiler is a C/C++ compiler, and the mbed Microcontroller simply runs a raw binary suitable for the target MCU. Therefore, existing ARM application code or middleware should be portable to mbed with a little care.

### **Q. Can I use the mbed compiler to compile programs for my own board?**

A. Yes, there is nothing special in the binary images generated by the compiler, so you can use it to write programs for your own designs that include the NXP LPC1768.

### **Q. What platform will mbed work on?**

A. As the microcontroller is a standard USB flash disk, it will work on Windows, Mac and Linux. The web based tools will run in Internet Explorer, Firefox, Safari and Chrome.

**Q. Are the mbed libraries open source?**

A. No. For now they are closed source, with just the compiled library and headers made available. This is to prevent fragmentation. They may become open source in the future, but not for now.

**Q. I already have a full license for ARM RealView/Keil MDK. Will mbed be useful?**

mbed is optimized for Rapid Prototyping, so it can be a useful tool regardless of experience, and alongside other tools.

It is certainly enabling users with skills outside embedded design to experiment, but for an experienced engineer it can still be an effective way to create proof-of-concepts and explore design spaces. It is likely they may then turn to tools optimized for implementing a fully productized solution afterwards, but mbed will have helped them create the right specification. Analogy: An architect still uses pencil and paper to sketch ideas before turning to an advanced CAD package to implement those ideas.

**Q. Will the mbed code be compatible with other compilers for production?**

The mbed Libraries are EABI and CMSIS compliant, and the mbed code is C/C++, so it will be possible to migrate to other toolchains. We'd like to work on making this a smooth process to help those that want to productize a design and go to volume production.

**Q. How does mbed relate to CMSIS?**

The mbed Libraries take advantage and are built on top of CMSIS (the Cortex Microcontroller Software Interface Standard). CMSIS provides a low level hardware abstraction layer for Cortex-M processors, whilst mbed provides a high level API.

**Q. Are all LPC1700 and LPC2300 peripherals supported by mbed?**

The mbed Library supports a set of common MCU peripheral APIs (Analog/Digital IO, PWM, Timers, Serial, SPI, I2C, CAN, Ethernet); all of these are supported on the LPC1768 and LPC2368. For the interfaces and options that are not supported by the mbed APIs, the peripherals can be accessed directly via CMSIS. The mbed Library is focused on providing an API-driven approach to working with peripherals, rather than exposing the hardware implementation details of the MCU.

**Q. Does mbed support the USB and Ethernet functions? Are stacks provided? Is it possible to use my own 3rd-party stacks?**

The mbed Library supports Ethernet within the API, and there are a number of USB, TCP/IP and HTTP contributed library examples. You are certainly free to incorporate 3rd-party or open source stacks, and we are working with middleware vendors to make this process very smooth.

**Q. Will mbed support any Real Time Operating Systems (RTOS) like FreeRTOS?**

There is currently no official support for RTOSes within mbed, although we expect unofficial ports to start appearing soon. We are also working with the RTOS vendors looking to support new customers, by integrating their RTOS solutions in to the mbed tools.

**Q. Are there any "extension boards" available in which to plug the mbed hardware?**

There are various extension boards appearing, such as a brain for an existing sumo robot, a Power-over-Ethernet breakout board, an adaptor for an existing modular development system. Anyone is free to create an extension board, and is encouraged to share the design for others. For anyone wanting to make an official extension board, you can contact [partners@mbed.org](mailto:partners@mbed.org) to discuss it further.

**Q. Am I required to share my code with the mbed community?**

There are certainly no requirements to share your code with the mbed community! However, we hope for many people the mbed community will present a great opportunity to help each other learn, share ideas and collaborate.

**Q. Can I sell my application code on mbed's community?**

There is currently no explicit support for selling code within mbed, but nothing prohibits it. We're working with middleware vendors on how we might enable this, and anyone interested can contact [partners@mbed.org](mailto:partners@mbed.org) to discuss it further.

**Q. Will mbed work with the upcoming Cortex-M0 LPC1100 series from NXP? What about other Cortex-M3 series like the new LPC1300?**

mbed currently supports the NXP LPC1768 Cortex-M3 MCU, and all of the LPC176x and LPC175x series members are similar. Additionally, NXP has good commonality among device peripherals, so can choose the optimal LPC1000 variant (LPC176x, LPC175x, LPC1300, etc) after developing your prototype. The LPC1100 Cortex-M0 series will also share similar peripherals to the other Cortex-M3 LPC1000 microcontrollers, so much of the prototype code you develop will be able to be migrated to the LPC1100 series.

**Q. Will mbed help me prototype a project for the LPC175x (80-pin) series?**

The LPC1768 has the same peripheral set as the LPC175x series. The prototype code you develop with the mbed NXP LPC1768 board will be fully applicable to your LPC175x production design, as long as you keep in mind the reduced number of general purpose I/O and peripherals. The mbed tool still allows you to create rapid prototypes to demonstrate proof-of-concepts and explore new design concepts.

**Q. I want to design to consume the lowest power possible. Will mbed help me?**

It is not something we have focused on, so it is probably not interesting for specific power optimization for existing ARM Cortex MCU users. But if mbed opens up the opportunity to use these microcontrollers for someone previously using older 8/16-bit designs or even discrete logic, it could be a real step change.