2.1 Introducing the mbed

Chapter 1 reviewed some of the core features of computers, microprocessors and microcontrollers. Now we are going to apply that knowledge and enter the main material of this book, a study of the ARM mbed.

In very broad terms, the mbed takes a microcontroller, such as we saw in Figure 1.4, and surrounds it with some very useful support circuitry. It places this on a conveniently sized little printed circuit board (PCB) and supports it with an online compiler, program library and handbook. This gives a complete embedded system development environment, allowing users to develop and prototype embedded systems simply, efficiently and rapidly. Fast prototyping is one of the key features of the mbed approach.

The mbed takes the form of a 2 inch by 1 inch (53 mm by 26 mm) PCB, with 40 pins arranged in two rows of 20, with 0.1 inch spacing between the pins. This spacing is a standard in many electronic components. Figure 2.1 shows different mbed views. Looking at the main features, labeled in Figure 2.1b, we see that the mbed is based around the LPC1768 microcontroller. This is made by a
company called NXP semiconductors, and contains an ARM Cortex-M3 core. Program download to the mbed is achieved through a universal serial bus (USB) connector; this can also power the mbed. Usefully, there are five light-emitting diodes (LEDs) on the board, one for status and four that are connected to four microcontroller digital outputs. These allow a minimum system to be tested with no external component connections needed. A reset switch is included, to force restart of the current program.

The mbed Architecture

The remainder of the pins connect to the mbed peripherals. These are almost all the subject of later chapters; we will quickly overview them here, though they may have limited meaning to you now.
There are no fewer than five serial interface types on the mbed: I²C, SPI, CAN, USB and Ethernet. Then there is a set of analog inputs, essential for reading sensor values, and a set of PWM outputs useful for control of external power devices, for example DC motors. While not immediately evident from the figure, pins 5 to 30 can also be configured for general digital input/output.

The mbed is constructed to allow easy prototyping, which is of course its very purpose. While the PCB itself is very high density, interconnection is achieved through the very robust and traditional dual-in-line pin layout.

Background information for the mbed and its support tools can be found at the mbed home page (Reference 2.1). While this book is intended to give you all information that you need to start work with the mbed, it is inevitable that you will want to keep a close eye on this site, with its cookbook, handbook, blog and forum. Above all else, it provides the entry point to the mbed compiler, through which you will develop all your programs.

2.1.1 The mbed Architecture

A block diagram representation of the mbed architecture is shown in Figure 2.2. It is possible, and useful, to relate the blocks shown here to the actual mbed. At the heart of the mbed is the LPC1768 microcontroller, clearly seen in both Figures 2.1 and 2.2. The signal pins of the mbed, as seen in Figure 2.1c, connect directly to the microcontroller. Thus, when in the coming chapters we use an mbed digital input or output, or the analog input, or any other of the peripherals, we will be connecting directly to the microcontroller within the mbed, and relying on its features. An interesting aside to this, however, is that the LPC1768 has 100 pins, but the mbed has only 40. Thus, when we get deeper into understanding the LPC1768, we will find that there are some features that are simply inaccessible to us as mbed users. This is, however, unlikely to be a limiting factor.

There is a second microcontroller on the mbed, which interfaces with the USB. This is called the interface microcontroller in Figure 2.2, and is the largest integrated circuit (IC) on the underside of the mbed PCB. The cleverness of the mbed hardware design is the way which this device manages the USB link and acts as a USB terminal to the host computer. In most common use it receives program code files through the USB, and transfers those programs to a 16 Mbit memory, which acts as the ‘USB disk’. When a program ‘binary’ is downloaded to the mbed, it is placed in the USB disk. When the reset button is pressed, the program with the latest timestamp is transferred to the flash memory of the LPC1768, and program execution commences. Data transfer between interface microcontroller and the LPC1768 goes as serial data through the UART (which stands for universal asynchronous receiver/ transmitter e a serial data link, let’s not get into the detail now) port of the LPC1768.
The LPC1768 Microcontroller

A block diagram of the LPC1768 microcontroller is shown in Figure 2.3. This looks complicated, and we do not want to go into all the details of what is a hugely sophisticated digital circuit. However, the figure is in a way the agenda for this book, as it contains all the capability of the mbed, so it is worth getting a feel for the main features. If you want to get complete detail of this microcontroller, then consult one or more of References 1.2, 2.3 and 2.4. Although these references are mentioned from time to time in the book, consulting them is not necessary for a complete reading of the book.
Remember that a microcontroller is made up of microprocessor core plus memory plus peripherals, as shown in Figure 1.4. Let’s look for these. Top center in Figure 2.3, contained within the dotted line, is the core of this microcontroller, the ARM Cortex-M3. This is a compressed version of Figure 1.6, the M3 outline that was considered in Chapter 1.

To the left of the core are the memories: the program memory, made with Flash technology, is used for program storage; to the left of that is the static RAM (random access memory), used for holding temporary data. That leaves most of the rest of the diagram to show the peripherals, which give the microcontroller its embedded capability. These lie in the center and lower half of the diagram, and reflect almost exactly what the mbed can do. It is interesting to compare the peripherals seen here with the mbed inputs and outputs seen in Figure 2.1c. Finally, all these things need to be connected together, a task done by the address and data buses. Clever though they are, we have almost no interest in this side of the microcontroller design, at least not for this book. It is sufficient to note that the peripherals connect through something called the advanced peripheral bus. This in turns connects back through a bus interconnect called the advanced high-performance bus matrix, and
from there to the central processing unit (CPU). This interconnection is not completely shown in this diagram, and we have neither need nor wish to think about it further.

**References**


*Part 2 of this excerpt continues with Getting Started with the mbed.*

**Copyright © 2014 Elsevier, Inc. All rights reserved.**

**Printed with permission from Newnes, a division of Elsevier. Copyright 2014. For more information on this title and other similar books, please visit www.newnespress.com.**