Chapter 1: Getting Started

HOW MANY MICROCONTROLLERS DID YOU USE TODAY?

A microcontroller is a kind of miniature computer brain that you can find in all kinds of devices. Some common, every-day products that have microcontrollers inside are shown in Figure 1-1. If it has buttons and a digital display, chances are it also has a programmable microcontroller brain.

Figure 1-1
Many devices contain microcontrollers

Try counting how many devices with microcontrollers you use in a day. If you hit your alarm clock's snooze button a few times in the morning, the first thing you did is interact with a microcontroller. Heating up some food in the microwave oven and making a call on a mobile phone also involve interacting with microcontrollers. Each of those microcontrollers is doing several jobs at once; the radio calculates the time, displays the numbers, tunes to the right station and reacts when you hit the snooze button. All of those devices have microcontrollers inside them that interact with you.

THE PROPELLER ACTIVITY BOARD – YOUR NEW EMBEDDED SYSTEM

Parallax Inc.'s Propeller Activity Board shown in Figure 1-2 has a multicore microcontroller built onto it; it is the largest black chip just above the Propeller Activity Board label. That chip has eight cores inside that perform the actual computing functions. The rest of the parts on the board support the microcontroller by providing power, a USB connection to your computer, extra memory, and sockets for connecting other devices. When all of these parts work together they are called an *embedded computer system*. This name is almost always shortened to just "embedded system."

> **Explore the Board**. Learn more about each part by reading the Propeller Activity Board's product documentation, a free download from www.parallax.com/product/32910.

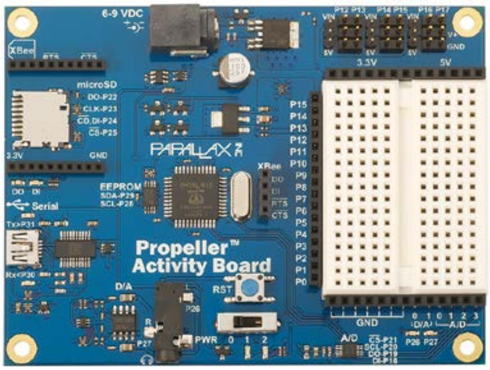


Figure 1-2
Propeller Activity Board with Built-in Propeller Microcontroller

The activities in this tutorial will guide you through building circuits with electronic parts similar to the ones found in consumer appliances and high-tech gadgets. You will also write computer programs that the Propeller chip will run. These programs will make the Propeller Activity Board monitor and control these circuits so that they perform useful functions.

AMAZING INVENTIONS WITH MICROCONTROLLERS

Consumer appliances aren't the only things that contain microcontrollers. Robots, machinery, aerospace designs, and other high-tech devices are also built with microcontrollers. Let's take a look at two in Figure 1-3 shows two robotic examples. On each of these robots, students use the Propeller microcontroller to read sensors, control motors, and communicate with other computers.

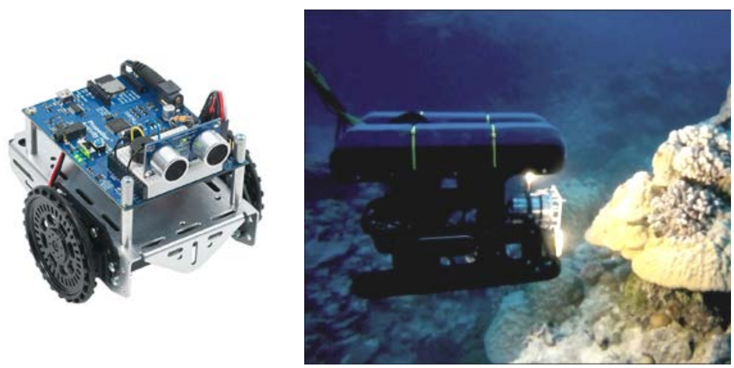


Figure 1-3
Educational Robots

ActivityBot robot (left)
CSU Monterey Bay Ulithi ROV Project (right)

The robot on the left is Parallax Inc.'s ActivityBot robot. It uses the Propeller Activity Board mounted on a small chassis with servo motors, wheels, and sensors to navigate by touch, visible light, infrared light, or ultrasound. The robot on the right is called an underwater ROV (remotely operated vehicle) and it was constructed at California State University of Monterey to study coral reefs in the Ulithi Atoll. Operators see what the ROV sees through a video camera feed, and control the ROV with a combination of hand controls and a laptop. Its Propeller microcontroller monitors sensors on the ROV and reports that information to the operator. At the same time, it also processes signals received from the operator's hand controls and relays them to the ROV's onboard motor controllers.

The flying quadcopter in the left of Figure 1-4 is called the ELEV-8. It was developed by Parallax, and its Propeller microcontroller manages the four motor-driven flight propellers so the aircraft remains stable and responsive to the operator's joystick controls. The millipede-like robot on the right of Figure 1-4 was developed by a professor at Nanyang Technical University, Singapore. It has more than 50 simple microcontrollers on board, and they all communicate with each other in an elaborate network that helps control and orchestrate the motion of each set of legs. In both of these vehicles, microcontrollers solve complex mechanical control problems. Robots not only help us better understand designs in nature, but they are being used to explore remote locations, disaster sites, and even other planets.

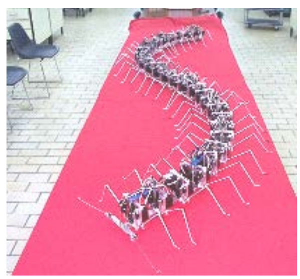


Figure 1-4
Research Robots

Parallax's ELEV-8 quadcopter (left) and Millipede Project at Nanyang University (right)

Microcontrollers are used in environmental applications, both unique and common. The weather station shown on the left of Figure 1-5 is part of a coral reef decay study. The microcontroller inside it gathers weather data from a variety of sensors and stores it for later retrieval by scientists. Even common traffic lights use sophisticated embedded systems to sense the presence of vehicles, coordinate with other lights to keep traffic moving smoothly, and detect preemption signals sent by emergency vehicle operators.

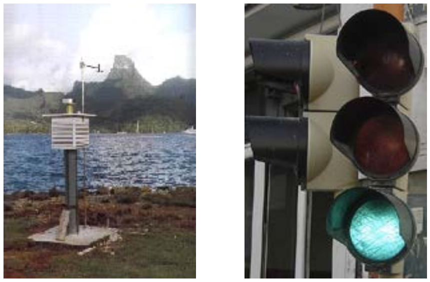


Figure 1-5
Environmental Devices

Ecological data collection by EME Systems (left); traffic light in Greece (right)

From your first project all the way through scientific applications, the microcontroller basics needed to get started on projects like these are introduced in this book. By working through the activities, you will get to experiment with a variety of building blocks like the ones found in all these inventions. You will build circuits for displays, sensors, and motion controllers. You will learn how to connect these circuits to the Propeller Activity Board's microcontroller, and then write programs that make it collect data from sensors, make decisions, and control lights or motion. Along the way, you will learn many important electronic and computer programming concepts and techniques. By the time you're done, you might find yourself well on the way to inventing a device of your own design.

ACTIVITY #1: WHAT'S A "MULTICORE" MICROCONTROLLER?

Okay, so now that we have seen where microcontrollers are used, what is a *multicore* microcontroller, and why would you want to use one? In each of the examples above, the microcontroller is doing more than one task at the same time. While a few tasks can be juggled at once by a single-core microcontroller, the more processes running at once, the more complicated it gets, especially with time-sensitive tasks like playing WAV files and controlling motors. Computers now come with dual, quad, and even eight cores to handle multiple tasks with speed and precision. Since a microcontroller is like a miniaturized computer that's designed to be the brains of products and inventions, it was inevitable that microcontrollers would also be designed with more cores.

The web article Propeller Brains for Your Inventions included below, breaks down and illustrates these concepts.

Propeller Brains for Your Inventions

The Propeller microcontroller on the Activity Board can be the brains for your own inventions, such as a robot. It is the brains of the ActivityBot robot, for example.

So, what is a microcontroller? It is an integrated circuit (computer chip) that includes a tiny processor to do the "thinking" and some memory so it can keep track of what it is doing. Microcontrollers also have input/output pins, **I/O pins** for short, which can exchange electrical signals with other devices such as lights, switches, beepers, motors, and sensors.

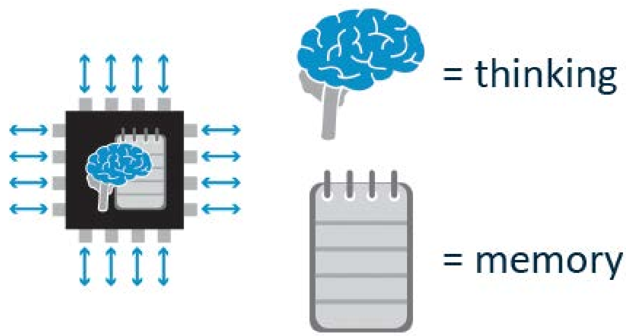


Figure 1-6
So What Is a Microcontroller?

A **single-core microcontroller** has just one processor inside. A **multicore microcontroller** has two or more processors, also called cores, inside one chip.

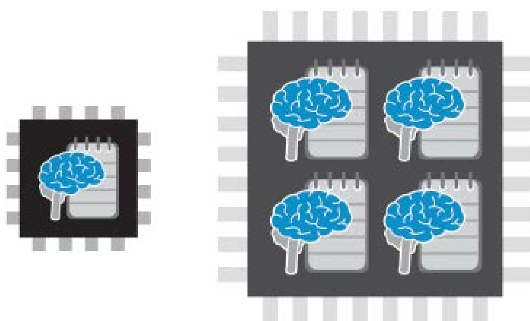


Figure 1-7
Single Core vs. Multicore

A single-core microcontroller is **multitasking** when it executes several tasks that must share its single processor. The processor must interrupt each task to switch briefly to another, to keep all of the processes going.

Imagine a chef in a kitchen alone, making bread, roast beef, and sauce. The chef must knead the bread dough for 15 minutes, interrupt that task every minute to stir the sauce, and remove the roast from the oven as soon as a thermometer reaches 120 °F. At any moment, the chef (processor) is executing only one task, while keeping all three processes (kneading, stirring, roasting) going at once.

Now imagine being that chef. The more tasks you must do at once, the more difficult it gets to keep track of them all, and keeping the timing right becomes more of a challenge.

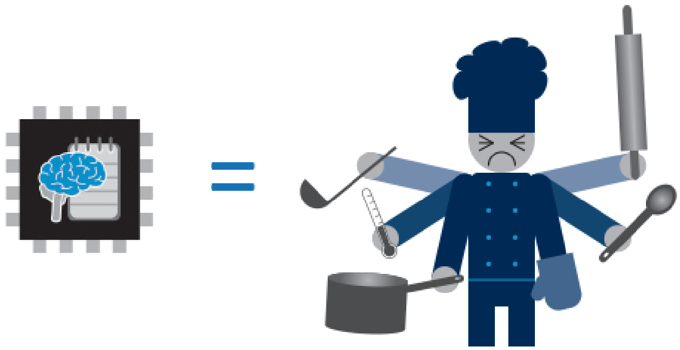


Figure 1-8
Multitasking

A multi-core microcontroller is **multiprocessing** when it executes several tasks at once, with each task using its own processor. This is also referred to as **true multitasking**.

Now, imagine a chef in a kitchen with three assistants, making bread, roast beef, and sauce. The chef puts one assistant at the stove to stir the sauce every minute. Another assistant is sent to keep watch on the thermometer, and remove the roast when it reaches 120 °F. Now the chef is free to knead bread dough for 15 minutes. The three cooks (processors) are keeping all three processes (kneading, stirring, roasting) executing at the same time, without any task-switching interruptions, and without missing the moment when the thermometer reaches 120 °F. There is even an extra assistant ready to help if something more is needed.

Having multiple cores makes it easier to do many tasks at once, especially if precise timing is needed.

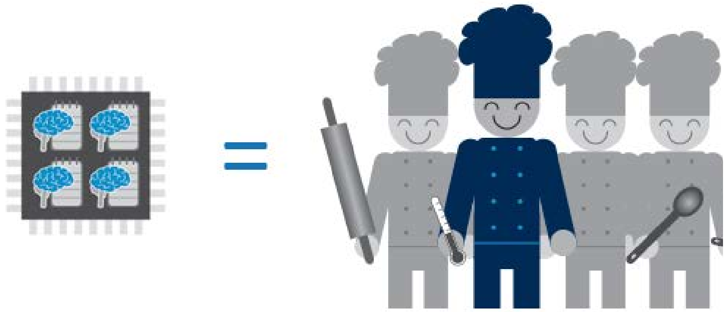


Figure 1-9
Multiprocessing

The Propeller microcontroller has 8 cores, and can therefore do multiprocessing, also called true multitasking. The cores are all the same. It has 32 I/O pins, which are also all the same. Each core can work with every I/O pin. This means that all of the Propeller cores and I/O pins are equally good at any tasks they must perform. Each core has a bit of its own memory. Each core also takes turn accessing a larger Main Memory, where they can share information.

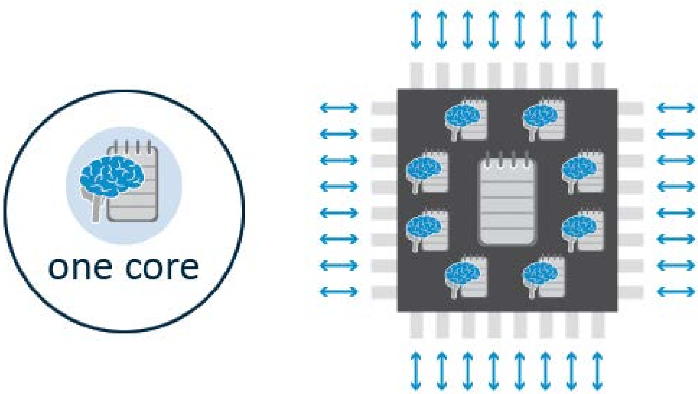
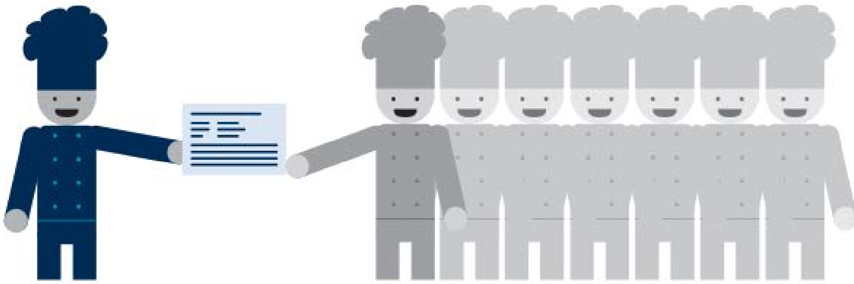


Figure 1-10
The Propeller Has 8 Cores

Eight cores in one microcontroller might sound intimidating. It might seem complicated to have to write programs for all of them. But the Propeller C language has pre-written code tasks, called **functions**, which make it easy.

Just think of functions as recipes the head chef can hand over to assistants, instead of having to explain to each one how to cook. One assistant can even ask another assistant for help, without bothering the head chef. Just as a team of 8 chefs can efficiently manage great meals, the Propeller with its eight cores can efficiently manage great inventions. Now that's teamwork!

Figure 1-11 Multicore Is Easy with C Functions



A C **library** is a collection of functions, sort of the way a cookbook is a collection of recipes.

Figure 1-12 A C Library is a Collection of Functions

So what does multicore processing look like for an invention like the ActivityBot?

Your C program might start a motor function, which makes another core manage the motors to make the robot move. Then, it might call a sensor function so the robot can "see" if there is an obstacle in its path. If an object is detected, your program then might call a music function, which will task other cores with the jobs of fetching songs from an SD card and playing music on an audio jack.

The ActivityBot is just one example of a microcontroller invention. You can use the Activity Board to build other projects or create your own inventions.