Building iPhone and iPad Electronic Projects

Why simply play music or go online when you can use your iPhone or iPad for some really fun projects, such as building a metal detector, hacking a radio-controlled truck, or tracking a model rocket in flight? Learn how to build these and other cool things by using iOS device sensors and inexpensive hardware such as Arduino and a Bluetooth Low Energy (LE) Shield.

This hands-on book shows you how to write simple applications with techBASIC, an Apple-approved development environment that runs on iOS devices. With just a few lines of code, you can write apps directly on your Apple device and have it interact with other hardware.

- Build a metal detector with the iOS magnetometer
- Use the HiJack hardware platform to create a plant moisture sensor
- Put your iPhone on a small rocket to collect acceleration and rotation data
- Hack a radio-controlled truck with Arduino and Bluetooth LE
- Create an arcade game with an iPad controller and two iPhone paddles
- Control a candy machine with an iOS device, a micro servo, and a WiFi connection

“Finally my iPad can interface with the real world! techBASIC is the easiest and most intuitive programming tool I have ever used. This book really makes me want to explore my creative ideas for controlling things with my iPad.”

—Jarle Bøe
Wireless Evangelist, Texas Instruments

Mike Westerfield runs Byte Works, Inc., an independent software publishing and consulting firm. He has developed iOS technical computing software, mission-critical software for military satellites, multimedia authoring tools for grade schoolers, and many other projects.
# Table of Contents

**Preface**........................................................................................................................................ vii

1. **Getting Familiar with techBASIC and Built-in Sensors.**.............................................................. 1
   - Your Own Tricorder .................................................................................................................. 1
   - A Crash Course in techBASIC ................................................................................................ 2
     - techBASIC Sampler ............................................................................................................. 2
     - Running Your First Program .............................................................................................. 3
     - Creating a Program ............................................................................................................. 5
   - The Accelerometer .................................................................................................................. 8

2. **Accessing the Other Built-in Sensors.** ...................................................................................... 21
   - The Gyroscope ...................................................................................................................... 22
     - Radians or Degrees? .......................................................................................................... 30
   - The Magnetometer ............................................................................................................... 30
   - Faster Sensor Response ........................................................................................................ 38
   - Heading .................................................................................................................................. 42
   - Location .................................................................................................................................. 43
   - Your Own Tricorder .............................................................................................................. 44

3. **Creating a Metal Detector.** ...................................................................................................... 45
   - The iPhone/iPad Magnetometer ............................................................................................ 45
   - The Earth's Magnetic Field .................................................................................................... 46
   - Using the iPhone or iPad as a Metal Detector ...................................................................... 48
     - Converting the Magnetometer Sample into a Metal Detector ........................................... 50
     - Using the Metal Detector .................................................................................................... 52
   - Finding Out More ............................................................................................................... 54

4. **HiJack.** .................................................................................................................................... 55
   - What Is HiJack? ..................................................................................................................... 55
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>Creating a Moisture Meter with HiJack</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>Adding a Moisture Meter to the Tricorder</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>Assembling the Moisture Meter</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>Calibration</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>Collecting the Calibration Data</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>Moving Datafiles to and from techBASIC</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>Using the Calibration Data</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Better Software</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>The Complete Moisture Meter Source</td>
<td>89</td>
</tr>
<tr>
<td>6.</td>
<td>Bluetooth Low Energy</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>What Is Bluetooth Low Energy?</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>The TI SensorTag</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>Writing Bluetooth Low Energy Programs</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>The Accelerometer</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>What’s an Accelerometer?</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>Accessing the Accelerometer</td>
<td>113</td>
</tr>
<tr>
<td></td>
<td>Using the Accelerometer</td>
<td>115</td>
</tr>
<tr>
<td></td>
<td>The Source</td>
<td>117</td>
</tr>
<tr>
<td></td>
<td>The Barometer</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>Accessing the Barometer</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>The Source</td>
<td>126</td>
</tr>
<tr>
<td></td>
<td>The Gyroscope</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>Accessing the Gyroscope</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>Using the Gyroscope</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>The Source</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>The Magnetometer</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Accessing the Magnetometer</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>Using the Magnetometer</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>The Source</td>
<td>141</td>
</tr>
<tr>
<td></td>
<td>The Humidity Sensor (Hygrometer)</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>Accessing the Hygrometer</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>The Source</td>
<td>147</td>
</tr>
<tr>
<td></td>
<td>The Thermometer</td>
<td>151</td>
</tr>
<tr>
<td></td>
<td>Accessing the Thermometer</td>
<td>151</td>
</tr>
</tbody>
</table>
You carry an amazing scientific instrument around in your pocket every day, using it for mundane tasks like making phone calls or listening to music. Your iPad 2 is as fast as a Cray-2 supercomputer from just a few decades ago, yet most people only use it to read books or surf the Web. What a waste.

This book is all about connecting your iPhone, iPod Touch, or iPad to the real world. You’ll start by learning how to access the sensors built right into your device. Next you’ll see how to connect wired sensors through the headphone port using a wonderful little device called HiJack. Several chapters show various ways to use Bluetooth low energy to connect to sensors, Arduino microcontrollers, motor controllers, and even other iPhones or iPads. Finally, you’ll see exactly how to use WiFi to connect to the Internet or physical devices connected to WiFi devices.

It would be pretty boring to make all of these connections just to make a few LEDs light up, so the book is organized around fun, interesting projects. The built-in sensors are used to create a metal detector. HiJack is hooked up to a simple electrical device so it can be used as a plant moisture sensor. Bluetooth low energy connects to a Texas Instruments SensorTag to detect acceleration to track the flight of a model rocket, and later to an Arduino microcontroller to hack a radio-controlled car, showing how to create robots and control them with your iPhone. Bluetooth low energy can also be used for peer-to-peer communication between iOS devices. You will learn how this is done by creating an arcade game that uses iPhones for game paddles. WiFi will be hooked up to a serial bridge to control servos, ultimately hacking a candy dispenser to give you candy under iPhone control.

Our look at each topic starts with a chapter that introduces the basic concepts using a simple project. One or more chapters follow these introductions, presenting the fun projects just mentioned. You may not want to build every one of them yourself, but reading through how they are created and how they work, you will get ideas about how to build your own projects.
You don’t need to go through this book linearly. If a project in the middle of the book seems really interesting, jump right to it. Each chapter starts with a section called “About This Chapter.” It lists the prerequisites, telling you which other chapters contain information you might need before attempting the project in the chapter you are interested in.

All of the hardware in the book is developed with electronic components you can buy from many Internet stores, but some of it is hard to find locally. Plan ahead. Glance at the parts list in a chapter a week or two before you want to get started, and order the parts you need.

Finally, the projects in this book cover several disciplines. There’s a lot of software, quite a bit of electronics, and a fair amount of mechanical engineering involved. Some of the stuff in this book is going to seem beyond your abilities. I know a few of the projects seemed that way to me as I wrote the book. After all, even though most of us have some technical ability, either through education or experience with hobbies, almost no one is fully qualified at computer science, electrical engineering, mechanical engineering, and physics.

Be brave, grasshopper.

Everything is laid out very carefully. If you don’t know much about software, start with the completely developed programs in the book, all of which are built right into tech-BASIC. If you don’t know one end of a battery from another, just wire stuff as you see it in the diagrams and photos that carefully document each circuit. As you learn more, you can experiment. Sure, there will be some failures along the way. I burned out a circuit or two and crashed a lot of software writing the book, and you’ll do the same as you read it. That’s how we learn.

I hope you don’t just build the projects in this book, though. The whole point is to learn how to do things, not just follow some plans. Whether you’re a professional trying to figure out how to remotely access data from a buried seismograph, a student exploring robotics for a science fair project, or an inventor tinkering with awesome ideas in your garage, I hope this book gives you some techniques and ideas that will enable you to create amazing things by combining software, electronics, and mechanics to build devices.

So, let’s go forth and control our world!

**Conventions Used in This Book**

The following typographical conventions are used in this book:

*Italic*

Indicates new terms, URLs, email addresses, filenames, and file extensions.
Constant width
Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

Constant width bold
Shows commands or other text that should be typed literally by the user.

Constant width italic
Shows text that should be replaced with user-supplied values or by values determined by context.

This icon signifies a tip, suggestion, or general note.

This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. Where this book includes code examples, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you’re reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O’Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product’s documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “Building iPhone and iPad Electronic Projects by Mike Westerfield (O’Reilly). Copyright 2013 James M. Westerfield, 978-1-449-36350-5.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.
Safari® Books Online

Safari Books Online is an on-demand digital library that delivers expert content in both book and video form from the world’s leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.


How to Contact Us

Please address comments and questions concerning this book to the publisher:

O’Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at http://shop.oreilly.com/product/0636920029281.do.

To comment or ask technical questions about this book, send email to bookquestions@oreilly.com.

For more information about our books, courses, conferences, and news, see our website at http://www.oreilly.com.

Find us on Facebook: http://facebook.com/oreilly

Follow us on Twitter: http://twitter.com/oreillymedia

Watch us on YouTube: http://www.youtube.com/oreillymedia
Acknowledgments

When I was a young nerd toting my slide rule back and forth to the library, one of my favorite books was *The Amateur Scientist*, a collection of articles from *Scientific American*. It was a remarkably diverse collection of projects. I added a significant amount of wear to that book, and eventually bought and wore out my own copy.

I hope this book is a lot like that one—it’s a book of projects, some of which you’re unlikely to take the time to build yourself. I hope you wear it out thumbing through the pages. As you do, though, keep in mind that it’s not the work of a single person. Oh, sure, I wrote it, but as Newton famously remarked, “If I have seen further it is by standing on the shoulders of giants.”

I owe a great deal to the people who educated me, both in and out of the classroom. A lot of them were in the early Apple II community. I won’t even try to name them, but you can find their footprints all through this book. Check out the KansasFest archives to meet some of these astoundingly creative people.

My wife is an amazing person. She’s my cheerleader, my critic, and the first person to read and correct each page. She watched our house as it was taken over by rockets, robot cars, and remote-controlled gadgets, encouraging me without complaining about the mess. She even pitched in on many of the projects. Among other things, the eyeball in Chapter 11 is her artwork. What an amazing best friend.

Thomas Schmid from the University of Utah took the time to answer a lot of questions about the HiJack, no doubt keeping me from frying a few. Like a lot of components, HiJack is manufactured by Seeed Studio. Leslie Liao from Seeed Studio kindly supplied the book’s reviewers with HiJacks so they could try the projects in Chapter 4 and Chapter 5.

I have some great new Internet friends at the Texas Instruments facility in Norway. Jarle Boe was fantastic, getting me started with the SensorTag before it even came out. He also let me use some of his photos, which are credited in the text. His staff was more than just helpful—Jomar Hoensi even wrote a special version of the firmware so it could collect data up to ±8G for rocket flights, and took the time to answer a lot of neophyte questions as I came up to speed on Bluetooth low energy. The rockets you see in Chapter 7 exist because of their efforts. I’m happy to say the rockets got to go to Norway for some trade shows, even if I never made it there myself.

My reviewers patiently slogged through all or part of this book. The amazing and talented Ryan family made up most of the reviewers. Kevin Ryan, Jess Finley, and Ken Moreland spent countless hours making sure everything worked and the descriptions were clear enough to follow. They even had electronics parties where they got together to build the projects. Doyle Maleche joined, from afar, bringing his experience as an educator to bear on the work. Kevin got me acquainted with a great O’Reilly author,
Alasdair Allan, who took the time to review parts of the book. Their comments made this a much better book than it would otherwise have been.

I’ve done a lot of writing for magazines over the years, and published software with a number of companies. While this is my first traditional book, I’ve worked with publishers and editors for a long time. I was pretty lucky to get some early training and encouragement from the editors and writers at Call A.P.P.L.E. I had pretty much given up on finding a publisher that really cared that much about its authors and products, but O’Reilly sure seems to be another one. I’ve been fortunate to have two great editors on this book. Brian Jepson got me started, then handed me off to Courtney Nash when Make: split from O’Reilly. Finding two people of their quality in a row says a lot for this company. If you decide to write, be sure to drop them a line. They are good people.

So, to all of you, from the Apple II buds in my early years to my newest friends at O’Reilly, thanks for making me look good!

Credits

While modified for the book, Chapter 5 originally appeared in the June 2012 issue of Nuts & Volts Magazine and is reprinted by permission of T & L Publications, Inc.

The SensorTag photo from Chapter 6 is courtesy of Jarle Bøe at Texas Instruments.

The illustration of the declination of the Earth’s magnetic field in Chapter 3 is courtesy of Wikimedia Commons.
CHAPTER 1
Getting Familiar with techBASIC and Built-in Sensors

About This Chapter

Prerequisites
You should already be familiar with using your iPhone. You should have some idea what programming is all about, although you do not need to be an ace programmer. It helps to know some variant of BASIC, but this is not required.

Equipment
You will need an iPhone, iPod Touch, or iPad running iOS 5 or later.

Software
You will need a copy of techBASIC or techBASIC Sampler.

What You Will Learn
This chapter starts with an introduction to techBASIC, the technical programming language used in this book for accessing sensors. It shows how to access the accelerometer that is built into every iOS device, starting with a simple one-line program and working up to a sophisticated accelerometer app.

Your Own Tricorder
I was always a little jealous when Spock pulled out his tricorder on Star Trek and began measuring practically every physical value you could imagine. It’s staggering how far technology has come, though. I carry a tricorder around in my pocket all the time now! Mine measures acceleration, rotation, and magnetic fields, giving both the strength and direction of each. It’s not quite as sophisticated as Spock’s, but it’s also not so large and clunky.
This book is all about using your iPhone and iPad to control electronic devices, often sensors. We’re going to start off with the sensors that are built right in, so you can pop out your tricorder and measure stuff, too.

**The iPod Touch**

The iPod Touch is essentially an iPhone without the phone, or, depending on your viewpoint, an iPad with a pocket-sized screen. Nothing in this book uses the phone part of the iPhone, so any time you see the iPhone mentioned, you can also use an iPod Touch. For the most part, we won’t talk about the iPod Touch specifically, but keep in mind that you can always use one instead of an iPhone.

**A Crash Course in techBASIC**

We’ll get started on the first instrument for our tricorder in a moment. First, though, let’s take a look at the language we’ll use for programming.

Our programs will be developed in a technical programming language called techBASIC, available in the App Store. There are a number of reasons for using techBASIC instead of Objective C, the programming language used to write most apps (including techBASIC). Here are some of the big ones:

- techBASIC runs right on your iPhone or iPad. You don’t have to use, or even own, a Macintosh computer to write or run these programs.
- techBASIC is less expensive. While Xcode (the development environment for Objective C) is free, you must join Apple’s developer program to actually move programs to your iOS device. That costs $99 every year. techBASIC costs $14.99 one time.
- techBASIC is simpler. It’s designed specifically for writing technical programs and connecting to external devices. Programs that would take a half-dozen to a dozen classes, each with a half-dozen to a dozen methods, can often be written with just a few lines in techBASIC.
- techBASIC is easier to learn and more forgiving than Objective C, so you can concentrate on the fun part—writing the programs to control the Internet of Things.

**techBASIC Sampler**

There is a free version of techBASIC called techBASIC Sampler. It’s also displayed in some places with the shortened name techSampler. The free version lets you view and run all of the samples that come with techBASIC. All of the programs from this book...
are samples in techBASIC, so you can use the free version to run the programs. You can even use the debugger to trace through the programs. The only limitation is editing. techBASIC Sampler doesn’t let you change a program or create a new one. When you try to edit a program or create a new one, techBASIC Sampler will invite you to upgrade to the full version of techBASIC using an in-app purchase. You can do that or just buy techBASIC.

That said, there are a couple of places in the book where we will create very short programs that are not preloaded as samples. These are typically used to show a simple feature before it gets buried in a longer program or to show how to use techBASIC. You can skip entering those programs without missing anything significant.

We’ll just talk about techBASIC in the book, but other than editing, you can always perform the same tasks with techBASIC Sampler.

Where to Get techBASIC

Like all iOS apps, techBASIC and techBASIC Sampler are available from Apple’s App Store.

- **techBASIC** is the full version of the development environment. It includes the samples from this book.
- **techBASIC Sampler** (also called techSampler, so the name will show up under the icon on the iPhone and iPad) is the free version of techBASIC. It allows you to run programs, including the samples from this book, but you cannot edit existing programs or create new ones. There is an in-app purchase to enable editing, which makes this program feature-for-feature compatible with techBASIC.

You can find more information about both programs, as well as a technical reference manual, at the Byte Works website.

Running Your First Program

Crank up techBASIC and you will see a display something like Figure 1-1, depending on the device you are using. If you are using an iPad and holding it in portrait view, tap the Programs button at the top left of the screen to see the list of programs. The iPhone will start off showing the programs, but if you switch to another display, you can switch back by tapping the Programs button at the lower left of the iPhone display.
About the Author

Mike Westerfield started programming on a PDP-8 using a teletype terminal. As the personal computer revolution got going, he sold his car and rode a bike for several months to raise cash to buy an Apple II computer. He wanted to write a chess program but couldn't find a good assembler, so he took a summer off to write his own. Two years later he finished ORCA/M, which went on to become Apple Programmer’s Workshop, the Apple development environment for the Apple IIGS.

Born the same year as Steve Jobs and Bill Gates, Mike made the mistake of getting an education instead of getting rich. A slow learner, he graduated from the US Air Force Academy in 1977 with a degree in physics, earned an MS in physics from the University of Denver, and was working on a PhD when he started making more money from his sideline software company than he did from the Air Force.

Since then Mike has developed numerous compilers and interpreters, software for mission-critical physics packages for military satellites, plasma physics simulations for Z-pinch experiments, multimedia authoring tools for grade schoolers, disease surveillance programs credited with saving the lives of hurricane Katrina refugees, advanced military simulations that protect our nation’s most critical assets, and technical computing software for iOS.

Mike currently runs the Byte Works, an independent software publishing and consulting firm. He is a PADI scuba instructor who lives in Albuquerque with his wife, where he enjoys being an empty nester and spoiling his grandchildren.

Colophon

The animal on the cover of Building iPhone and iPad Electronic Projects is a Eurasian magpie (Pica pica). The magpie is a very common bird throughout Europe, much of Asia, and northwestern Africa. Although considered a pest by some due to its penchant for eating small songbirds, the magpie is one of the smartest species of birds, and indeed one of the smartest animals in general. In the wild, magpies have been observed engaging in elaborate social rituals; they use tools, hide and store food, and employ complex group hunting strategies. Other members of the magpie’s family, corvidae, include crows and ravens, and all of these species have exhibited mirror self-recognition in captivity. Some primates and cetaceans share this capability; these animals have a large brain-to-body weight ratio that is only slightly lower than that of humans. It has even been shown in laboratory tests that the brains of corvidae birds have evolved the same ability to think geometrically as the great ape’s.

Wild magpies form monogamous pairs after attending large gatherings that Charles Darwin described as “marriage meetings.” A mating pair of magpies will stay together for their entire lives, raising broods of five to eight chicks every year. The chicks stay in the nest for a few weeks after hatching and still remain with the parents for about a week
after learning to fly. It is thought that this long period of adolescence helps contribute to the birds’ intelligence, since the chicks have ample time to learn social behaviors from their parents.

The name “magpie” is derived from a mention in Shakespeare’s *Hamlet* of a “magot pie,” or “pied Margot.” A pied (or piebald) animal is one that has a spotting pattern of large white areas over black; in the play, Hamlet describes this bird along with “choughs and rooks” who bring forth “the secret’st man of blood.” The magpie has long been entwined with European folklore, and even in Shakespeare’s time the bird was looked upon as a dark omen. Most of the myths associated with the magpie are a result of the bird’s tendency to “steal” shiny objects or of its aggressive behavior toward songbirds. In some areas of Britain, it is still traditional to greet a solitary magpie with “Hello, Mr. Magpie, how is your wife today?” to ward off misfortune.

The cover image is from Wood’s *Natural History*. The cover font is Adobe ITC Garamond. The text font is Adobe Minion Pro; the heading font is Adobe Myriad Condensed; and the code font is Dalton Maag’s Ubuntu Mono.