Extracted from:

Programming Your Home

Automate with Arduino, Android, and Your Computer

This PDF file contains pages extracted from *Programming Your Home*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The Pragmatic Programmers

Programming Your Home

Automate with Arduino, Android, and Your Computer

Mike Riley

Edited by Jacquelyn Carter

Pragmatic Bookshelf

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at *http://pragprog.com*.

The team that produced this book includes:

Jackie Carter (editor)
Potomac Indexing, LLC (indexer)
Molly McBeath (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Printed in the United States of America.
ISBN-13: 978-1-93435-690-6
Printed on acid-free paper.
Book version: P1.0—February 2012

Welcome to the exciting, empowering world of home automation! If you have ever wanted your home to do more than just protect you against the outside elements and want to interface it to the digital domain, this book will show you how. By demonstrating several easy-to-build projects, you will be able to take the skills you learned from this book and expand upon and apply them toward custom home automation projects of your own design.

The book's primary objective is to get you excited about the broader possibilities for home automation and instill the confidence you need to ultimately build upon these and your own ideas. The projects also make great parent-child learning activities, as the finished products instill a great sense of accomplishment. And who knows? Your nifty home automation creations may even change the world and become a huge new business opportunity for other homeowners actively seeking an automation solution that saves them time and money.

Who Should Read This Book

Programming Your Home is best suited to DIYers, programmers, and tinkerers who enjoy spending their leisure time building high-tech solutions to further automate their lives and impress their friends and family with their creations. Essentially, it is for those who generally enjoy creating custom technology and electronics solutions for their own personal living space.

A basic understanding of Arduino and programming languages like Ruby and Python are recommended but not required. You will learn how to combine these technologies in unique configurations to resolve homemaker annoyances and improve home management efficiencies.

In addition to the inclusion of Python scripts and Ruby on Rails-based web services, several of the projects call upon Google's Android platform to help enhance the data event collection, visualization, and instantiation of activities. A basic familiarity with the Android SDK will be beneficial so that the projects that make use of the Android OS can offer a more mobile reach.

If you're the type of person who prefers to build versus buy your home accessories, this book will further motivate you to use what you learned in the book as a starting point to expand upon and optimize them in various ways for their environment. Even though some of the topics deal with multiple software- and hardware-based solutions, they are easy to follow and inexpensive to build. Most of all, they show how a few simple ideas can transform a static analog environment into a smart digital one while having fun.

What's in This Book

After a basic introduction to home automation and the tools of the trade, this book will teach you how to construct and program eight unique projects that improve home utility and leisure-time efficiencies. Each project incorporates a variety of inexpensive sensors, actuators, and microcontrollers that have their own unique functions. You will assemble the hardware and codify the software that will perform a number of functions, such as turning on and off power switches from your phone, detecting package deliveries and transmitting emails announcing their arrival, posting tweets on Twitter when your bird feeder needs to be refilled, and opening and closing curtains depending on light and temperature, and more.

Because the recommended skill set for building these solutions includes some familiarity with programming, this book builds upon several previously published Pragmatic Bookshelf titles. If you would like to learn more about programming Arduinos or writing Ruby or Python scripts, I strongly recommend checking out the books listed in Appendix 2, *Bibliography*, on page ?.

Each project begins with a general introduction and is followed by a What You Need section that lists the hardware parts used. This is followed by a section called Building the Solution that provides step-by-step instructions on assembling the hardware. *Programming Your Home* will call upon the Arduino extensively for most (but not all) of the projects. Once the hardware is constructed, it can be programmed to perform the automation task we built it to do. Programs can range from code for Arduino microcontrollers to scripts that execute on a computer designed to control, capture, and process the data from the assembled hardware elements.

The book concludes with a chapter on future projections in home automation and a chapter filled with idea starters that reuse the hardware and software approaches demonstrated in the eight projects.

Arduinos, Androids, and iPhones, Oh My!

With the meteoric rise of mobile device proliferation, the post-PC moniker has made its way into the tech world's vocabulary. I am a big proponent of technology shifts, but I am also old enough to have lived through three major computing revolutions (the shift from mainframes to PCs, the rise of the Internet, and the shift from PCs to mobile devices) and know that change isn't as fast as people say it is. Until mobile applications can be developed on mobile devices the way PC applications can be developed on PCs, a Linux, Windows, or Mac computer will be a central requirement for developing mobile apps. The same holds true for Arduino programming.

That said, the times are indeed a-changing. Microsoft Research was one of the first major phone OS providers to attempt to create native mobile applications directly on the mobile device with their release of TouchStudio. Google engineer Damon Kohler created the Scripting Layer for Android (SL4A) that gives Android users the ability to write fairly sophisticated programs using a text editor on their phone. Coupled with Sparkfun's IOIO ("yo-yo") board, we're already seeing early glimpses of what could replace the PC for some of the scripts created for this book.

Since you will need a Mac, Linux, or Windows computer to program the Arduinos and mobile apps in this book, this computer will also be the machine that runs the server-side programs that interpret and extend information out to your mobile devices. Of course, if you only have one computer and it's a laptop that travels with you, consider purchasing a cheap Linux or Mac to run as your home server. Not only will you benefit from having a dedicated system to run the monitoring apps 24/7/365, but it can also serve as your home Network Attached Storage (NAS) server as well.

I am a believer in open source hardware and software. As such, the projects in the book depend upon these. I am also technology-agnostic and rarely have any overriding devotion to one hardware supplier or programming language. Code for this book could have been presented just as easily in Mono-based C# and Perl, but I opted for Ruby and Python because of their portability and multiparty open source support. I could have used a Windows or Linux machine as the server and development system but chose Mac for the book because Ruby and Python are preinstalled with the OS, thereby eliminating the time and space required to install, configure, and troubleshoot the operating environment.

In accordance with this open source philosophy, I also opted to demonstrate the mobile application examples exclusively for the Android OS. While I personally prefer iOS devices as the platform of choice for my mobile lifestyle, the overhead associated with writing applications for iOS is a hassle. In addition to learning Objective-C and the various frameworks as well as dealing with the burden of memory management, deploying iOS apps requires either a jailbroken device or the legitimate purchase of an annual membership to Apple's iPhone developer network. Conversely, Android's SDK and application deployment is free and open. Android programs can also multitask better than iOS programs. Of course, these two advantages also bring on greater security and resource utilization risks. That said, I encourage readers who prefer the mobile demos to run on non-Android devices to port the simple client programs presented in this book to their favorite mobile OS and share these conversions with the Programming Your Home community.

Another term that is gaining a foothold in the tech press is the "Internet of Things." This phrase refers to the idea that with the proliferation of network-connected microcontrollers, Internet-based communication between such small devices will eventually outnumber people surfing the Web. While that may be the case for submitting data upstream, reaching such a device from the Internet is still a hassle. Besides the technical knowledge required to set up a dynamic DNS and securely configure port forwarding to easily reach the device, ISPs may block outbound ports to prevent end consumers from setting up dedicated servers on popular network ports like FTP, HTTP/S, and SMTP.

The projects in this book should work perfectly fine in a home local area network. However, obtaining sensor data outside of this local network is a challenge. How do you check on the status of something like a real-time temperature reading without going through the hassles of opening and forwarding ports on your router (not to mention the potential security risks that entails)?

Fortunately, several companies have begun to aggressively offer platforms accessible via simple web service APIs to help overcome these hassles. Three of these gaining momentum are Pachube, Exosite, and Yaler.[1] Configuring and consuming their services is a fairly straightforward process. I encourage you to visit these sites to learn more about how to incorporate their messaging capabilities into your own projects.

Code Examples and Conventions

The code in this book consists of C/C++ for Arduino, Java for Android, Ruby for web middleware, and Python for desktop scripts. Most of the code examples are listed in full, except when burdened by external library overhead (such as in the case of Android and Ruby on Rails program listings). Syntax for each of these languages is highlighted appropriately, and much of the code is commented inline along with bullet markings to help bring attention to the big ideas in the listings.

Highlights and sidebars are used sparingly in the book in an effort to keep information moving at a quick yet manageable clip.

[1] http://www.pachube.com, http://www.exosite.com, and http://www.yaler.org, respectively.

Online Resources

Visit the book's website at http://pragprog.com/titles/mrhome, where you can download the code for all the projects, participate in the book's discussion forum, ask questions, and post your own home automation ideas. Bugs, typos, omissions, and other errors in the book can be found on the book's errata web page.

Other popular website resources include the popular DIY websites Makezine, and Instructables,[2] where participants share a wide variety of home-brewed creations with their peers.

There are also several IRC channels on freenode.net and SIG forums on Google Groups dedicated to the subject, with many focused on singular aspects of DIY gadget design, home automation, and hardware hacking.[3]

OK, enough with the preamble. Let's get ready to build something!

[2] http://www.makezine.com and http://www.instructables.com, respectively.

[3] http://groups.google.com/group/comp.home.automation/topics