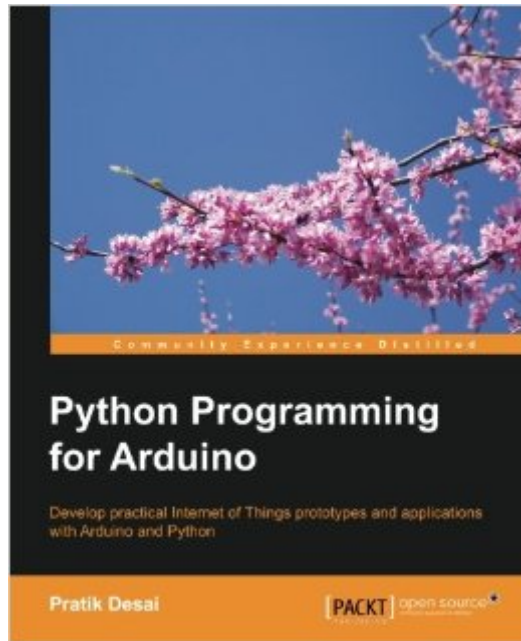


The book was found

Python Programming For Arduino



Synopsis

Develop practical Internet of Things prototypes and applications with Arduino and Python About This Book Transform your hardware ideas into real-world applications using Arduino and Python Design and develop hardware prototypes, interactive user interfaces, and cloud-connected applications for your projects Explore and expand examples to enrich your connected device's applications with this step-by-step guide Who This Book Is For This is the book for you if you are a student, hobbyist, developer, or designer with little or no programming and hardware prototyping experience, and you want to develop IoT applications. If you are a software developer or a hardware designer and want to create connected devices applications, then this book will help you get started. What You Will Learn Design and develop your own hardware prototypes using Arduino Interface Arduino to a computer using the Firmata protocol and Python Acquire hands-on experience in developing applications for a variety of popular sensors Develop graphical user interfaces to control your components and plots to visualize sensor data Access your Arduino prototype from the Internet Implement messaging protocols for Arduino communication Connect your hardware project to cloud-based services Create advanced Internet of Things applications through projects in the book In Detail The future belongs to applications and services that involve connected devices, requiring physical components to communicate with web-level applications. Arduino combined with the popular open source software platform Python can be used to develop the next level of advanced Internet of Things (IoT) projects with graphical user interfaces and Internet-connected applications. Starting with designing hardware prototypes using Arduino, this book will then show you everything you need to know to be able to develop complex cloud applications. You will delve into domain-specific topics with incremental complexity, ending with real-world projects. You will quickly learn to develop user interfaces, plots, remote access, messaging protocols, and cloud connectivity. Each successive topic, accompanied by plenty of examples, will help you develop your cutting-edge hardware applications.

Book Information

Paperback: 400 pages

Publisher: Packt Publishing (February 27, 2015)

Language: English

ISBN-10: 1783285931

ISBN-13: 978-1783285938

Product Dimensions: 7.5 x 0.9 x 9.2 inches

Shipping Weight: 1.9 pounds (View shipping rates and policies)

Average Customer Review: 4.8 out of 5 stars [See all reviews](#) (17 customer reviews)

Best Sellers Rank: #824,923 in Books (See Top 100 in Books) #61 in [Books > Computers & Technology > Hardware & DIY > Mainframes & Minicomputers](#) #321 in [Books > Computers & Technology > Hardware & DIY > Single Board Computers](#) #604 in [Books > Computers & Technology > Programming > Languages & Tools > Python](#)

Customer Reviews

The book begins with an appropriate introduction to three main subjects: Python programming language, the dialect of C/C++ used in Arduino, and the basics of the Arduino platform. Those subjects are covered in the first 47 pages and are not explained in depth, but all the main features are properly introduced. Anyone who is familiar with a high level language can find this jump start suitable for his needs. The book is full of interesting tools and tips to speed up your learning experience. I particularly appreciate that there are many examples of code; first of all in Python, but often also in Arduino's own language. The next part of the book provides a lot of recipes for various components (servo and DC motors, LEDs, speakers and so on) presenting a mini-project with the appropriate hardware configuration and a suitable Python program to deal with it. If you need to find the proper hardware to complete those mini-projects, you will be pleased to notice that there is a very useful choice of URLs to online stores (such as [Amazon](#)). Then, the author describes how to build graphical interfaces with a GUI library called Tkinter, storing and reading data from files and plotting charts with input values from sensors. At the end of this part, a more complex example is introduced: DIY Thermostat. To complete this project, Raspberry PI is described and there are the main guidelines to use it to deploy the project. Another half of the book is devoted to illustrate the use of Arduino and Python dealing with networks. At the beginning the basics are presented; then the author deals with "Internet of Things" projects and very advanced features such as web servers, REST interfaces and AWS services.

First my background: I'm an old software/hardware guy who first used the Intel 4004 microprocessor back in 1973. Since then I have been professionally involved with microprocessors and microcontrollers, even after retirement. I've taught college level courses on the micro. I know and have used quite a few languages from assembly to C++. Going all the way through this book would keep anyone out of trouble for a long time. I must admit not getting all the way through at this writing, although I have skimmed the remainder. The author, in his easy to follow style, is honest and points

out when Python should be used and when other "static"(compiled) languages such as C would be better choice. He demonstrates use of both C and Python in the examples. Fair treatment of the three major operating systems is refreshing. Those are Windows, Mac, and Linux. Linux may not be the most popular, but many experimenters who would find this book useful do use Linux. He also spends a good amount of time bringing new users of Python and the Arduino gently up to speed. Some background in writing computer programs and elementary knowledge of electronics is desirable, but may not be necessary as he explains most things in detail. The "Firmata" protocol was totally foreign to me, and I finally figured out (correctly?) that the more common term might be "debugger". First time encountering that term, but of course "sketch" (meaning source code) was also foreign to me before encountering the Arduino. I did not see the common term "bootloader" mentioned. The author is using the terminology of the Arduino enthusiast rather than those of mainstream hobbyists and professionals.

I had an Arduino Uno board at home for some time. I had few projects in my head but I didn't have the required knowledge of electronics and C programming to bring them to an end. I have been programming in Python for a long time and I bought this book with the hope of being able to somehow simplify the projects by using a more familiar programming language. The book was extremely useful as it clearly explains which part of the code should be written in C and when it is possible to use Python. The first chapter of the book introduces the reader to both Python and Arduino and can be safely skipped if the reader is already familiar with both of them. The author provides detailed step by step instructions on how to install what is needed. Throughout the book instructions are given for Linux, Mac OS X, and Windows. The next chapter is about the Arduino Firmata protocol and the pySerial Python library. This chapter was probably one of the most interesting to me as it shows how to interact with the Arduino board by using Python. The following chapters are a little bit more complex but they are still easy to follow, and the author shows how to use various electronic components such as a passive infrared (PIR) sensor to detect movements, a potentiometer, a buzzer, a DC motor, a servomotor, a temperature sensor, a humidity sensor, and a light sensor. Almost every chapter proposes little projects that can be as easy as blinking an LED or rather complex. The instructions of the projects are always very clear and the provided schemes are (at least for me that I have no experience in electronics) extremely useful. A portion of the book is devoted to provide a GUI to the projects. The author chose to use Tkinter.

[Download to continue reading...](#)

Python: PYTHON CRASH COURSE - Beginner's Course To Learn The Basics Of Python

Programming In 24 Hours!: (Python, Python Programming, Python for Dummies, Python for Beginners, python crash course) Python: Learn Python In A DAY! - The Ultimate Crash Course to Learning the Basics of Python In No Time (Python, Python Course, Python Development, Python Books, Python for Beginners) Programming #45: Python Programming Professional Made Easy & Android Programming In a Day! (Python Programming, Python Language, Python for beginners, ... Programming Languages, Android Programming) Python: Learn Python FAST - The Ultimate Crash Course to Learning the Basics of the Python Programming Language In No Time (Python, Python Programming, ... (Learn Coding Fast with Hands-On Project 7) PYTHON: Python in 8 Hours, For Beginners, Learn Python Fast! A Smart Way to Learn Python, Plain & Simple, Learn Python Programming Language in Easy Steps, A Beginner's Guide, Start Coding Today! Programming Raspberry Pi 3: Getting Started With Python (Programming Raspberry Pi 3, Raspberry Pi 3 User Guide, Python Programming, Raspberry Pi 3 with Python Programming) Python: Learn Web Scraping with Python In A DAY! - The Ultimate Crash Course to Learning the Basics of Web Scraping with Python In No Time (Web Scraping ... Python Books, Python for Beginners) Arduino: Complete Beginners Guide For Arduino - Everything You Need To Know To Get Started (Arduino 101, Arduino Mastery) Programming #8:C Programming Success in a Day & Android Programming In a Day! (C Programming, C++programming, C++ programming language, Android , Android Programming, Android Games) Programming #57: C++ Programming Professional Made Easy & Android Programming in a Day (C++ Programming, C++ Language, C++for beginners, C++, Programming ... Programming, Android, C, C Programming) Programming: Computer Programming for Beginners: Learn the Basics of Java, SQL & C++ - 3. Edition (Coding, C Programming, Java Programming, SQL Programming, JavaScript, Python, PHP) Arduino: The Ultimate QuickStart Guide - From Beginner to Expert (Arduino, Arduino for Beginners) Hacking: Hacking Made Easy 1: Beginners: Python: Python Programming For Beginners, Computer Science, Computer Programming Swift and Python Programming Guide: Programming Language For Beginners: Learn in a Day! Box Set Collection (Swift, Python, JAVA, C++. PHP) Python: Learn Python in One Day and Learn It Well. Python for Beginners with Hands-on Project. (Learn Coding Fast with Hands-On Project Book 1) Python Programming for Arduino Raspberry Pi 2: Raspberry Pi 2 Programming Made Easy (Raspberry Pi, Android Programming, Programming, Linux, Unix, C Programming, C+ Programming) Android: Programming in a Day! The Power Guide for Beginners In Android App Programming (Android, Android Programming, App Development, Android App Development, ... App Programming, Rails, Ruby Programming) DOS: Programming Success in a Day: Beginners guide to fast, easy and efficient learning of DOS programming (DOS, ADA, Programming, DOS

Programming, ADA ... LINUX, RPG, ADA Programming, Android, JAVA) ASP.NET: Programming success in a day: Beginners guide to fast, easy and efficient learning of ASP.NET programming (ASP.NET, ASP.NET Programming, ASP.NET ... ADA, Web Programming, Programming)

[Dmca](#)