Make: Getting Started with Intel Edison

No bigger than a postage stamp and priced at just $50, the tiny but mighty Intel Edison is comparable in power to an entry-level Windows 8 tablet. Makers love it for its versatility as a communications device, for its applications to wearable technology, and for its ability to connect with the world.

Running a Linux-based operating system and Arduino compatible, the Edison is Internet of Things-ready and wants to be the brains of your next project!

Written by an Intel research scientist, this book introduces the Edison, explains its many possibilities, and lays out the foundation for a beginning Maker to start using it immediately.

With Make: Getting Started with Intel Edison, you’ll learn to:

- Use Computer Vision tools to recognize faces and objects
- Create your own MP3 player and audio recorder
- Recognize and decipher human speech—make your own Siri or Echo!
- Code in Python on the Edison
- Connect external devices with I2C and SPI
- Install additional software packages with opkg

This book introduces you to the wide world of possibilities that the tiny Edison contains. You’ll find complete installation instructions, an introduction to the Arduino-based programming environment, and a guide to getting started with Python programming on the Edison. You’ll also discover how to work with sensors, add Bluetooth to your project, and connect to the Internet with WiFi.

Start small, make big!
Make: Getting Started with Intel Edison

Stephanie Moyerman
## Contents

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

**1/Introduction to Edison.** ........................................... 1
   Tour of Breakout Boards. ........................................... 2
      The Arduino Breakout Board. ................................... 3
      Other Breakout Options. ........................................ 6
   Setup and Configuration. .......................................... 11
   Install. ..................................................................... 13
      Mac and Windows. ................................................ 13
   Connecting. ............................................................ 16
      Windows. ................................................................ 16
      Mac. ..................................................................... 19
      Linux. .................................................................... 19
   Logging In. .............................................................. 20
      Configuring Edison and Getting Online. .................... 20
   Troubleshooting. ....................................................... 23
   Going Further. .......................................................... 24

**2/Introduction to Linux.** ............................................. 25
   What Is Linux? ......................................................... 25
   The Edison Filesystem. ............................................... 26
   Basic Linux Commands. ............................................. 29
      Accounts, Permissions, and Ownership. .................... 35
   Scripting and More Advanced Linux Commands. ............ 38
   The Internet. ............................................................ 41
   Installing Packages in Yocto. ...................................... 44
   Text Editors. ............................................................ 46
   Going Further. .......................................................... 47

**3/Introduction to Arduino.** ......................................... 49
   What is Arduino. ....................................................... 49
   Materials List. .......................................................... 50
   The Arduino IDE. ...................................................... 51
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing.</td>
<td>51</td>
</tr>
<tr>
<td>Navigating the IDE.</td>
<td>52</td>
</tr>
<tr>
<td>Sketches and Functions.</td>
<td>55</td>
</tr>
<tr>
<td>Digital Output with Blink.</td>
<td>57</td>
</tr>
<tr>
<td>Going Further with Blink.</td>
<td>59</td>
</tr>
<tr>
<td>The Blink Circuit.</td>
<td>61</td>
</tr>
<tr>
<td>Digital Input: Adding a Button.</td>
<td>68</td>
</tr>
<tr>
<td>The Serial Console.</td>
<td>70</td>
</tr>
<tr>
<td>Toggling the LED.</td>
<td>73</td>
</tr>
<tr>
<td>Analog Output.</td>
<td>75</td>
</tr>
<tr>
<td>Analog Input.</td>
<td>78</td>
</tr>
<tr>
<td>I2C Accelerometer.</td>
<td>82</td>
</tr>
<tr>
<td>SPI Screen.</td>
<td>87</td>
</tr>
<tr>
<td>Linux, C++, and the Arduino IDE.</td>
<td>92</td>
</tr>
<tr>
<td>Troubleshooting.</td>
<td>95</td>
</tr>
<tr>
<td>Going Further.</td>
<td>95</td>
</tr>
<tr>
<td>4/Programming in Python.</td>
<td>97</td>
</tr>
<tr>
<td>Introduction.</td>
<td>97</td>
</tr>
<tr>
<td>“Hello, World” in Python.</td>
<td>98</td>
</tr>
<tr>
<td>Functions and Loops.</td>
<td>101</td>
</tr>
<tr>
<td>Installing Libraries.</td>
<td>102</td>
</tr>
<tr>
<td>Blink in Python.</td>
<td>103</td>
</tr>
<tr>
<td>Scripting at Bootup.</td>
<td>105</td>
</tr>
<tr>
<td>Button-Controlled Blink.</td>
<td>106</td>
</tr>
<tr>
<td>Bluetooth-Controlled LED.</td>
<td>106</td>
</tr>
<tr>
<td>Bluetooth Pairing.</td>
<td>107</td>
</tr>
<tr>
<td>Exchanging Information.</td>
<td>109</td>
</tr>
<tr>
<td>The Potentiometer.</td>
<td>112</td>
</tr>
<tr>
<td>I2C Accelerometer.</td>
<td>113</td>
</tr>
<tr>
<td>Installing the Dependencies.</td>
<td>115</td>
</tr>
<tr>
<td>Using the MMA Library.</td>
<td>115</td>
</tr>
<tr>
<td>SPI Screen.</td>
<td>118</td>
</tr>
<tr>
<td>BLE Beacon.</td>
<td>120</td>
</tr>
<tr>
<td>Edison Side.</td>
<td>120</td>
</tr>
<tr>
<td>Smartphone Side.</td>
<td>122</td>
</tr>
<tr>
<td>Going Further.</td>
<td>123</td>
</tr>
<tr>
<td>5/Teach Edison to See.</td>
<td>125</td>
</tr>
<tr>
<td>Introduction.</td>
<td>125</td>
</tr>
<tr>
<td>Materials List.</td>
<td>126</td>
</tr>
</tbody>
</table>
Intel Edison is ... well, it’s hard to say. Intel Edison is so many different things. If someone told you about Edison 10 years ago, you would have thought that person was utterly insane.

At its core, Intel Edison is a very powerful and cheap little computer. Priced at around $50 USD, this dual-core Atom processor is comparable to an entry-level Windows 8 tablet. At only a small fraction of the cost, it contains all the software amenities of modern-day computing. The system contains 1 GB of memory and 4 GB of storage to handle heavy computational tasks and data logging.

Intel Edison is a communications device. It comes integrated with Bluetooth and WiFi capabilities and preinstalled software to run both.

Intel Edison is a hardware development platform. It contains an Intel Quark microcontroller that allows you to program and control connected devices and circuits. It’s also Arduino-compatible, meaning the vast majority of shields, code examples, and libraries that have been specifically designed for Arduino will work as is with Edison, too.

Finally, Intel Edison is an embedded device, designed for the Internet of Things (IoT) and wearable technology. Even though Intel Edison is a computer, communications device, and hardware development platform, it weighs only eight grams, is approximately the size of a postage stamp, and can run in low power modes.

While on the subject, it’s worth taking a moment to discuss what Edison is not. Although Edison is a standalone computer
running a full Linux operating system, it’s meant to be the brains of your connected and wearable devices—not a laptop or desktop in the traditional sense. For this reason, Edison does not interface with a display, keyboard, or mouse on its own; you connect to Edison through a host computer and load your software directly on the module. Edison is not a great choice if you’re looking to build a home media console or old-school arcade game (though both could still be done, with a lot of work), but it’s a great candidate for a personal assistant, smart watch, automated robot, smart-home controller, or basically any other electronic system you can imagine. What will you make?

---

**Total Cost**

The Edison module alone costs around $50, and if you add the accompanying components to complete every exercise in this book, the total cost is around $200. For a complete materials list, see Appendix A.

---

**What You Can Do with It**

As you can see, Intel Edison is a versatile platform that can be used for a great many applications:

*Learn about computers*

Edison is a great and inexpensive way to learn more about computers at the lowest level. With Edison, you can learn Linux, configure your operating system, learn about kernels and builds, and install and program drivers for your accessories. You can learn about WiFi and Bluetooth networking by programming Edison to be a dynamic wireless hotspot or Bluetooth beacon. Chapter 2 provides an introduction to Linux and the specific operating system (OS) on Edison. Chapter 3 and Chapter 4 touch on Bluetooth commands.

*Learn to program*

Edison is an amazing tool for learning to program; it comes preloaded with many different compilers and interpreters,
and installing more is a breeze. From Chapter 4 onward, this book focuses mainly on programming in Python, but Edison also supports Java, node.js, C, C++, and many more languages. Even Arduino programming is expanded with Edison. Standard Arduino programs consist of compiled C++ based on the `avrlibc` library, but Edison exposes the Arduino IDE to the full C++ standard programming environment. With these additional libraries, you can make system calls and tap into the power of Linux within your Arduino sketches, making Arduino for Edison a powerful tool. You’ll see how in Chapter 3.

Make
You can use Edison to program your electronics projects. Edison and its accompanying ecosystem support the same inputs and outputs (I/Os) as the Arduino Uno and integrate seamlessly with almost any existing Arduino project (more on this in Chapter 1 and Chapter 3). With its size, connectivity, and computing power, you can build elaborate and computationally heavy systems that interface directly with the web or store data on board. In Chapter 3, you’ll create a range of electronic circuits powered by Edison and program them using the Arduino IDE. Because programming electronics on Edison is not limited to the Arduino IDE, you recreate these same projects in Chapter 4 by programming them in Python. After that, you’ll interface some of these electronics with other computational tasks, leveraging the full power of Edison.

Go to market
Edison was created specifically to lower the barrier to entry for makers hoping to take their ideas and prototypes all the way to product. Edison interfaces with a variety of breakout boards (discussed in Chapter 1), allowing you to rapidly prototype on one while resting assured that your final design will work on another. In this way, you can develop and program without worrying about the final form factor; Edison will handle this for you.
What’s Been Done with It

Sometimes it’s hard to decide what to do with such an interesting new device, especially one with such a range of options. Although Edison has been, at the time of this writing, in production for under a year, a wealth of amazing projects have already been created and you can look to for inspiration:

- Intel’s project gallery
- Hackster’s gallery
- Intel’s Make It Wearable finalists
- Hackaday.io

Who This Book Is For

This book is an introductory tutorial for Intel Edison. It is meant to showcase the versatility of the product and therefore spans a wide range of topics, from Linux to hardware interfaces to Python programming. You don’t need experience in any of these topics to get started, just a little curiosity and a desire to build new things. The only thing you’ll need are some basic computer skills: the ability to move and manipulate files, search within your computer, and install software.

The aim of this book is to help you get started designing, building, and programming end-to-end systems with Edison. This book is in no way a comprehensive guide to systems engineering or computer science—there are many resources available should you want to delve deeper into either of those topics. Instead, this book is meant to inspire you, so that you can take your ideas and concepts very quickly to reality.

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 element signifies a tip, suggestion, or a general note.

This element indicates a warning or caution.

Using Code Examples
This book is here to help you get your job done. In general, 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 MAKE 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: Make: Getting Started with Intel Edison by Stephanie Moyer-Man (Maker Media). Copyright 2016, 978-1-4571-8759-9.
How to Contact Us

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

Make:
1160 Battery Street East, Suite 125
San Francisco, CA 94111
877-306-6253 (in the United States or Canada)
707-639-1355 (international or local)

Make: unites, inspires, informs, and entertains a growing community of resourceful people who undertake amazing projects in their backyards, basements, and garages. Make: celebrates your right to tweak, hack, and bend any technology to your will. The Make: audience continues to be a growing culture and community that believes in bettering ourselves, our environment, our educational system—our entire world. This is much more than an audience, it’s a worldwide movement that Make: is leading—we call it the Maker Movement.

For more information about Make:, visit us online:

Make: magazine: http://makezine.com/magazine
Maker Faire: http://makerfaire.com
Makezine.com: http://makezine.com
Maker Shed: http://makershed.com

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at: http://bit.ly/gsw-intel-edison.

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

Acknowledgements

There are a great many people I want to acknowledge for helping me, either directly or indirectly, with this book. First, the Intel Edison team, for supplying me with enumerable Edisons to play with over the past 12 months. Special thanks to Ed Ross, Jim
Chase, and Jay Melican for putting up with me during that time (and hopefully moving forward, too).

Second, to my bosses, Jeff Ota and Lakshman Krishnamurthy, who give me the time and freedom to play with Edison as a part of my actual job. To the remainder of the team, who challenge me every day to build something cool and do something with my life—thank you guys, too.

Super special thanks to my technical editors, Jason Wright (two thank yous!) and Esther Kim, for just being amazing people and friends and putting up with the many edits that this book needed. To Jonathan and Julija for playing the role of my non-technical editors and working through all the examples in the book (and also putting up with my needed edits).

Finally, thanks to my parents for letting me be a very curious child and indulging my need to take everything apart. And to my beautiful wife Kelsey, for tolerating my endless string of electronic and coding projects and my tendency to leave half-finished projects all over our otherwise clean house. I love you.