

EDUCATION

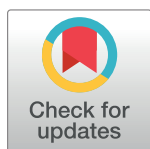
Ten quick tips for using a Raspberry Pi

Anthony C. Fletcher^{1*}, Cameron Mura²

1 Salix Management Consultants Ltd, London, United Kingdom, **2** Dept of Biomedical Engineering, University of Virginia, Charlottesville, Virginia, United States of America

* drtonyfletcher@gmail.com

This is a *PLOS Computational Biology* Education paper.



Overview

Much of biology (and, indeed, all of science) is becoming increasingly computational. We tend to think of this in regard to algorithmic approaches and software tools, as well as increased computing power. There has also been a shift towards slicker, packaged solutions—which mirrors everyday life, from smart phones to smart homes. As a result, it’s all too easy to be detached from the fundamental elements that power these changes and to see solutions as ‘black boxes’. The major goal of this piece is to use the example of the Raspberry Pi, a small, general-purpose computer, as the central component in a highly developed ecosystem that brings together elements like external hardware, sensors and controllers, state-of-the-art programming practices, and basic electronics and physics, all in an approachable and useful way. External devices and inputs are easily connected to the Pi, and it can in turn control attached devices very simply. So, whether you want to use it to manage laboratory equipment, sense and sample the environment, teach bioinformatics, control your home security, or make a model lunar lander, it’s all built from the same basic principles. To quote Richard Feynman, ‘What I cannot create, I do not understand’.

OPEN ACCESS

Citation: Fletcher AC, Mura C (2019) Ten quick tips for using a Raspberry Pi. *PLoS Comput Biol* 15(5): e1006959. <https://doi.org/10.1371/journal.pcbi.1006959>

Editor: Francis Ouellette, University of Toronto, CANADA

Published: May 2, 2019

Copyright: © 2019 Fletcher, Mura. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: Portions of this work were supported by the University of Virginia and US National Science Foundation CAREER award MCB-1350957. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Introduction

Recent Raspberry Pi sales figures [1] show that this humble board is the world’s third-best-selling general-purpose computer, after personal computers (PCs) and Macs, with over 15 million sold. (Various brand names are used in the examples in this work; there are several competitors to most of them, and we are not favouring one over the other. Rather, we use them simply as examples of a wider group; for instance, ZDnet lists 12 competitors to a Pi [2].) Although it may be a fair bet that many Pi units are sitting unused in dusty cupboards, equally many of them are beavering away at a multitude of practical tasks, at home and work, day in, day out, without blinking. (They do blink; they have a light-emitting diode [LED]!) One of us (ACF) has one he hasn’t rebooted or touched in over a year. It just does its work. What is the Pi, and why should anyone care?

For those of you new to this realm, the Pi is a small computer made and sold by the trading arm of a United Kingdom-based charity [3] whose initial aim was to ‘promote the study of computer science and related topics, especially at school level, and to put the fun back into learning computing’. A historical account of the Raspberry Pi Foundation’s early roots is offered in [4]. The Pi’s integrated circuit design is that of a system-on-chip (SoC), versus the

usual motherboard-based PCs familiar to most readers. A major architectural difference here is that the core components in an SoC-based microcomputer—e.g., the central processing unit (CPU)-housing microcontroller, memory blocks, voltage regulators, etc.—are built as a single, fully integrated circuit, enabling small chips with low power consumption, reduced heat dissipation, and so on. These features in turn have made such systems popular for embedded and mobile devices. The Pi uses an Advanced RISC Machine (ARM) chip, such as those found in about half of all smartphones [5], as opposed to the Intel/Advanced Micro Devices (AMD) chips found in most PCs. About as powerful as an iPhone 5, the Raspberry Pi 3 has a multi-core score of around 1300; detailed performance metrics and score information for the iPhone5 and Pi 3 can be found at [6] and [7], respectively. The Pi generally runs a special distribution of the Debian (Unix-based) operating system (OS) termed 'Raspbian' (Fig 1); other Linux distributions, as well as other (non-Linux) OSs, can be run too.

In practice, the Pi is a small computer (without a case) about the size of a bar of soap (Fig 1A). It features the usual universal serial bus (USB) ports and high-definition multimedia interface (HDMI) sockets, as well as one extra set of gizmos (which are what make the Pi so versatile), known as a set of 'general-purpose input-output' (GPIO) pins. The GPIO, composed of 40 pins, enables you to directly tell the Pi to send or read a voltage on them. This in turn enables you to plug in about a gazillion peripherals and read and control them directly, all without any soldering. Using a little device known as a 'breadboard' [11], you can also rapidly build and control basic electronic circuits. (Often used in prototyping circuits, the breadboard is a device to link resistors, transistors, and other components without soldering.)

This simplicity and modularity of the Pi is one of the reasons why it has become so popular: after a small learning curve, you aren't limited by your skills, only by your imagination. The Pi is a real jack of all trades, and it is manifestly not (just) a toy: for example, Los Alamos National Laboratory recently built a Raspberry Pi cluster with 3,000 cores as a pilot before scaling up to 40,000 cores or more [12,13].

A Pi shouldn't be confused with, for example, an Arduino. The latter, which is an open-source, single-board microcontroller, does not run an OS; instead, a microcontroller is designed to run relatively simple programs over and over, depending on the intended use (e.g., as a temperature or light sensor), often as an embedded system. The Arduino is quite useful but differs somewhat from the Pi in its range of applicability and intended uses; helpful comparisons of the Pi and Arduino can be found at [14,15].

So, the Pi is an inexpensive device—a minicomputer, basically—that can have a place in any lab, home, car, garden, and office. Here are ten quick tips for using one. Note that these tips are intentionally broad in scope so as to illustrate the general utility of this sort of device, including its role in the professional development of any scientist by virtue of developing one's computational skill set and knowledge of computer hardware principles. As a more concrete context, and in alignment with this journal's more typical readership, some explicitly biosciences-related examples are included in Tip 10.

Tip 1: Don't be put off by the Pi being incredibly inexpensive (in fact, there's an even cheaper version . . .)

For maybe \$60 USD—or less, if you already have items like a secure digital (SD) card or a suitable mobile-phone charger lying around—you have all that you need to get going. While a Pi Model B is around \$40 USD, you also need an SD card and a power supply (a phone charger will do at push, but a dedicated 2A supply is recommended). If you have a desktop computer, you can do what's called a "headless setup" with no further costs and operate the Pi via an approach like Virtual Network Computing (VNC) [16]. Given all of that, you can also simply

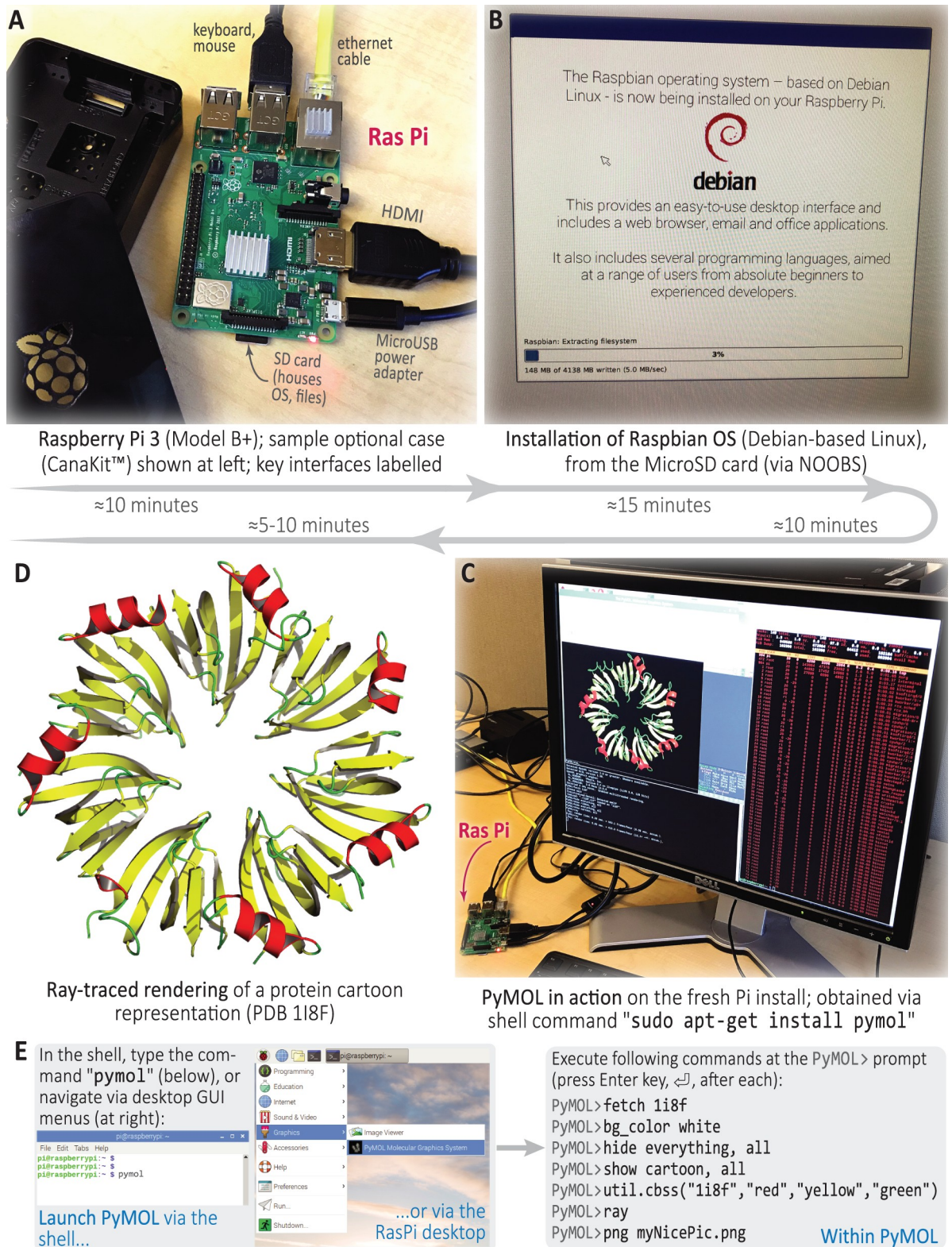


Fig 1. A 'real' scientific example: molecular visualization with PyMOL on the Pi in under an hour. This montage shows a Pi being used to create publication-quality molecular graphics. Starting from a pristine board (A), the Raspbian OS can be readily installed on, say, a 32 GB MicroSD card such as that shown in (B) via the NOOBS installation manager; NOOBS is available at <http://bit.ly/rasPiNOOB> (which resolves to the longer URL in [8]). Then, after installing PyMOL from the command line and spawning a session (C), a protein 3D structure

can be retrieved from the Protein Data Bank [9], loaded into PyMOL, and ray traced (D); the PyMOL-related steps are detailed in (E), and primers such as those in [10] can be consulted for further details. The approximate duration of each of the steps (A) → (D) is indicated alongside the panels. NOOBS, New Out Of Box Software; SD, secure digital.

<https://doi.org/10.1371/journal.pcbi.1006959.g001>

add a keyboard, mouse, and display to make the Pi a standalone unit. Note that the Pi could even pay for itself: just use one to monitor the energy usage in your house and turn the thermostat down a bit occasionally. And, if the kids knock a soda onto it, who cares—there are no moving parts. A smaller-sized ‘Pi Zero W’ is also available, which comes with built-in Wi-Fi and Bluetooth, and retails for about \$10 USD (however, minimally, you will need a power source and an SD card). Many individuals use a full-size Pi, such as a third-generation Raspberry Pi 3, to prototype their application and then simply run the production version on the smaller device (if the usual task/load isn’t too compute-intensive). The Pi Zero W is ideal for remote cameras and environment sensors and can work for considerable periods supplied only by a mobile-phone power pack. After a ‘coffee incident’, one of us (ACF) rinsed a Pi Zero W under a tap, and it worked fine once it had dried. (It was unplugged, and this is not recommended, but *c’est la vie*.)

Tip 2: Try it if you’re interested in more than just *using* a computer

Many of us who are scientists probably remember the thrill of discovering things for the first time as children. The Pi gives you the chance to relive such moments as an adult. We mentioned earlier the small learning curve to get a Pi up and running. Small it is, but it’s there, and it’s a challenge to overcome. Once you decide what you need, it takes only a few Google searches and you have access to the collective knowledge of many such enthusiasts and ‘maker’ communities [17,18]. Therein lies its charm: although the Pi is (intentionally) not slick, and it doesn’t ‘just work’ (like turning on a tablet and firing up an app), its configurability is virtually unlimited. You learn to do so much more with it—not just as an open-source PC but also as a ‘brain’ to control a multitude of peripherals (see, for example, those in Fig 2).

You can rediscover your old-school physics about what resistors, transistors, and capacitors do, and even if you don’t like coding, it’s still relatively straightforward to use the Pi, as there is a graphical user interface (GUI) and many prebuilt applications. If you are so minded, learning the basics of two languages (PHP, Python) and a smattering of Unix/Linux can get the device really singing. The Pi is like a kind of modern, digital, utilitarian jigsaw kit. All of the parts and code are out there, so you can just put them together to make something bespoke; unlike a jigsaw, Pi-based devices are much more than just decorative.

Tip 3: If you like repetitive work, don’t buy one

As with any computer, the Pi excels at automating tasks, and especially so, as it’s simple to connect to a wide range of sensors and external inputs. There’s always a toss-up between writing a little script to automate something versus just doing it manually (on occasion). How you decide to proceed—to script or not to script—is generally dictated by your personal balance of two countervailing forces: (i) the effort required of you (the cost of scripting) versus (ii) the frequency with which the task arises (often, or only intermittently?—this is the cost of not scripting). If you tend to like the idea of scripting/programming and automating, then you’ll probably enjoy the Pi from the get-go. Particularly with a shell script, or perhaps Python, R, or MATLAB, it becomes as quick as possible to use the Pi to automate processes. It’s very easy to add the Pi to your home or lab network, share drives, etc. (all behind a firewall of course), and simple, one-line scripts that run at regular intervals (‘cron jobs’, after the standard Unix command ‘cron’) can be written for repetitive tasks (for example, turning on the coffee pot at 6:00 each morning).

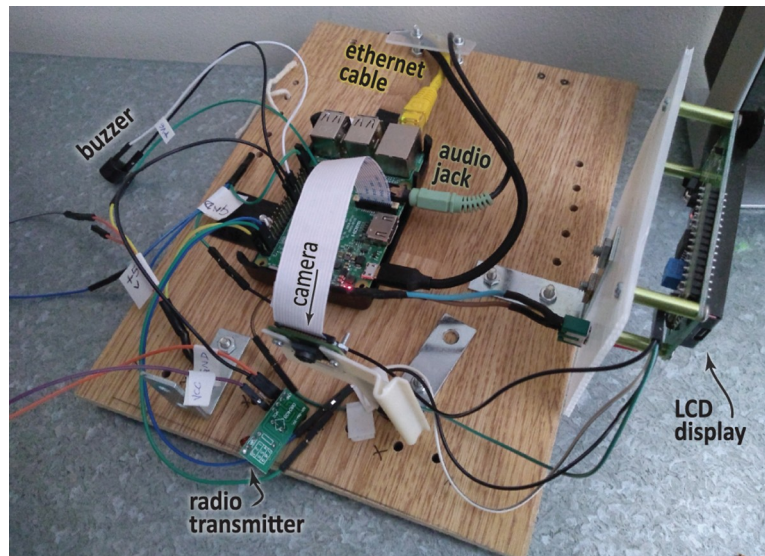


Fig 2. Developing your own Pi-based system. This annotated photograph shows a typical Pi in action—under development, with attached components that provide further input or output functionalities (LCD display, audio cable, camera, network cable, 433-MHz radio transmitter, and a buzzer). *Photo credit: AC Fletcher.* LCD, liquid crystal display.

<https://doi.org/10.1371/journal.pcbi.1006959.g002>

So, you can simply automate things that would be just too laborious (impossibly so) with a pen and paper or are otherwise unamenable to manual intervention. For example, during a recent hot spell, one of us (ACF) became interested in how the temperature in the house varied with the windows open and the blinds drawn. The measuring peripheral cost \$5 USD, and the code took about ten minutes. (Search your favourite auction site for a DHT22 or AM2302 temperature and humidity sensor or a BME280, which adds barometric pressure as well.)

Tip 4: Have fun

Have you ever wanted to turn the lights on when it gets dark without getting up? Have a radio-frequency identification (RFID) controller on a fridge to control who accesses it? See who is at your front door? Learn what is eating the nuts you left out for the squirrels? Shine a laser beam across the doorway so that you can log who comes in and out? (And maybe take a photo or high-definition (HD) video? For around another \$25 USD, you can get a very decent camera.) A Pi opens the door to all of these capabilities—all in addition to tasks like logging, temperature recording, basic artificial intelligence (AI) and ‘on-the-go’ data science, image recognition, text-to-speech conversion, and vice versa.

Essentially, you are limited only by what you want to do: from basic robotic kits to mechanical arms, there are a host of possibilities. And many of these activities and projects have been done already (or at least started), so you can simply Google them and use other work or build on people’s existing efforts (see also Tip 7).

Because of its simple interface with sensors, motors, cameras, etc., the Pi can be used for many practical tasks (see again Fig 2). Yesterday’s youngsters loved model railways and constructor sets, but today they might love to make motion sensor-based teddy bears that laugh when you wave your hand.

Just one example: using MIT’s App Inventor tool [19], it’s easy to make a little app that, when you say “play X” into an Android phone, tells the Pi plugged into the hi-fi to do just that. Now no one in the house (reading the paper in silence) is safe from a high-volume rendition of

Bobby Boris Pickett and the Crypt-Kicker Five blasting out ‘Monster Mash’. The Android-specific App Inventor is a cloud-based tool, so you can build mobile phone and tablet apps right in your web browser; other app-development platforms and utilities exist too (e.g., Thinkable [20] allows development of cross-platform apps, meaning they can operate on iOS as well as Android platforms).

Tip 5: Embrace open-source

Much of the software commonly used on a Pi is free of cost, and there are plenty of great tools to bring it all together. For example, the Pi’s open-source Raspbian OS, mentioned in the Introduction, is based on Debian, with all that that involves; indeed, much of the software and peripherals one uses with a Pi comply with free and open-source (FOSS) principles, meaning no restrictive licenses or costly fees. (For more on the issue of ‘free’, as in licensing and use [‘freedom’] versus cost [‘free pizza’], as well as the general issue of software licensing and FOSS, see [21].) It would be remiss of us not to expand on the exceptional MIT App Inventor tool. This is a visual programming environment that allows anyone—including even children—to build fully functional apps for smartphones and tablets. It’s brilliant with a Pi, as you can easily write basic ‘apps’ to interface your phone or tablet with your Pi and communicate over the internet or just your own local area network (LAN; basically, a home network). You can also, of course, run a basic web server on the Pi and communicate that way. It’s as easy as Pi. The world really is your oyster here.

Although not Pi-specific, many organisations and data suppliers (especially public utilities) also offer application programming interfaces (APIs) that enable you to ‘talk’ to them and get their data in a structured, parsable way. For example, one of ACF’s Pi’s automatically gets the local bus times and flashes up when the next bus is due so that he knows when to stroll down to the stop and not have to hang around.

And, in addition, you can give back to the community by sharing your code with others (which you’re encouraged to do, fear or anxieties aside [22]!). In Tip 7, we mention the repositories of data for the Pi; well, you can easily answer others’ questions here or even set up your own simple website to share your code (which of course can be hosted on your Pi!). People will certainly appreciate any and all sharing. More broadly, the Internet-in-a-Box project [23] is an example of a recent free and open initiative that employs the Pi as a hardware back-end to essentially provide all Wikipedia content (as well as other open content, e.g., of medical relevance) on a single physical unit; such devices can serve indispensable medical and educational roles in much of the world (e.g., imagine healthcare providers in rural regions or offline communities, or areas with various access/control restrictions and other hindrances to unfettered internet access).

Tip 6: Welcome an Internet of Things that isn’t going to disappear

The Internet of Things (IoT) [13,24] is already upon us and will only increase in its ubiquity; and it’s currently a standards nightmare. From a smart fridge to smart heating (well, smart anything), off-the-shelf solutions are generally quite expensive and, more crucially, closed-source and proprietary. So, your light bulbs might not talk to your central heating, and your front doorbell could struggle to communicate with your fridge. More importantly, if the manufacturer goes broke, you’ll be left with a lot of useless consumer gear. It doesn’t have to be like this, though: using open-source rather than proprietary technologies—e.g., open-source software libraries, modern programming languages, and standards-compliant tools like User Datagram Protocol (UDP) and radio frequency (Rf) communication for networking—you can easily communicate between devices and across LANs and the internet. It’s also simple to

converse with virtual-assistant devices such as Siri and Alexa via their APIs. More generally, your Pi now places you into the realm of IoT concepts like ‘edge’ and ‘fog’ computing [24].

You can be as sophisticated or as simple as you like—with a Pi Zero W at \$10 USD, it doesn’t break the bank to try things. Everyone will have a different set of needs. As well as some of the uses mentioned earlier, consider \$3 USD wireless switches that tell if a door or window opens or similarly priced passive infrared (PIR) sensors and a microwave sensor that you can wave at to turn on the mains power (relatively inexpensive at \$6 USD a pop). One of ACF’s next tasks is to set up a dedicated barcode scanner by the kitchen waste bin to remind him what to buy at the supermarket based on what he throws away.

Tip 7: Engage with online communities (they can be helpful and virtuous places, just beyond the trolls, fake news, and Twitter bots)

One of the great things about the Pi—and, in general, the sociology and culture that underlie Linux, Python, PHP, and other open-source efforts [25]—is that the folks who contribute to these communities typically have a genuine interest and desire to both share information and help others [26]. There is generally much prompt and free help, and opportunities for collaboration may even arise. Indeed, large communities online are dedicated to the Pi [27]; in a wider sense, you may already know Stack Exchange, GitHub, Quora, and others, but it is worth mentioning them in a little detail here.

Stack Exchange, launched in 2008, offers over 150 communities built by people passionate about a focused topic. These Q&A sites, comprehensively indexed at [28], range from hermeneutics, to a parenting site, to quantum computing. Many of these are computer-related communities or are otherwise scientific/technical in nature. For example, many Pi-related discussions can be found in a dedicated site [29], as well as in the broader Stack Overflow (which is the very first of the many Stack Exchange sites); other stacks also contain helpful information (e.g., molecular visualization on the Pi is discussed at [30]). You can just join and ask a question, and it may well be answered within minutes. It’s also worth carefully trawling through old posts: if you’re good at constructing Google queries, you can search something like ‘python iterate through list’, and you’ll get a ready-made chunk of code that allows you to go one by one down a list and do stuff. Indeed, such is the value of this open, community-based approach that BioStars, founded as an online bioinformatics resource, was modeled after Stack Exchange [31].

Similar in spirit, GitHub is an online resource where one can host code, manage projects using the state-of-the-art distributed version-control system known as Git [32], and build software alongside millions of other developers. This is a central place to perhaps collaborate on projects or maybe just download a precooked module to run a Pi peripheral—for example, buy a 4-digit LED display (as for a clock radio), Google search ‘Python TM1637’, and you’ll find yourself at the doorstep of all the device drivers that some kind soul has already written and tested. Finally, Quora is a popular site, too, modelled on a rich question-and-answer functionality and offering its own Pi-based communities [33].

Though sometimes a bit nerdy, you can’t enter the above communities and not feel the warmth of human sharing, endeavour, and altruism.

Tip 8: It’s versatile

With 15 million devices out there and peripherals costing literally pennies, it’s unsurprising that they find themselves everywhere. For a little inspiration, try the listicle ‘50 of the most important Raspberry Pi Sensors and Components’ [34].

The Pi is also used across a wide range of sciences, from the amateur to the professional. For example, a webcam has been controlled by a Pi to monitor fiddler crabs [35], and a device

known as the AirPi can record and upload information about temperature, humidity, air pressure, light levels, UV levels, carbon monoxide, nitrogen dioxide, and smoke level to the internet [36]. If you want to go into the chemistry lab, then how about an internet-based reaction monitoring, control and autonomous self-optimization platform for chemical synthesis [37]? And if you're interested in the environment more generally, have a look at Public Lab, a community in which you can learn how to investigate environmental concerns using inexpensive do-it-yourself (DIY) techniques [38].

Tip 9: You can do everyday science experiments with it for just pennies

A friend wanted to know the least windy spot on his balcony, as his plants were always getting blown over. A \$5 USD anemometer rotor, a quick Google search of the Hall effect, some off-the-shelf logging software, and the job was done. How can the Pi do this? The Hall effect is "the production of a voltage difference (the Hall voltage) across an electrical conductor, transverse to an electric current in the conductor and to an applied magnetic field perpendicular to the current" [39]. So, as the magnet moves past the coil you get a change in voltage, which the Pi can detect; it's good for measuring rotational speeds, as there are no moving parts that can cause mechanical switches to play up if the rotor is going really fast. (This didn't stop the friend's box hedge from getting infested with caterpillars, but you can't have everything.) Seriously though, most of the peripherals that you can get from online auction sites cost only a few dollars (yes, really), and although most of them are somewhat rudimentary (e.g., measure distance ultrasonically to a few cm, light levels to a few lux), often they are good enough for basic DIY experimentation.

The Pi's ability to simply log data makes it tremendously useful, with only minimal expertise required. The Pi's GPIO pins essentially read if a voltage is there or not (they can read the level of a voltage as well, to an extent), so as long as you can make whatever you are running emit an electrical signal of some kind, then the Pi can log it (and it can also easily log Rf signals with a tiny \$5 USD superheterodyne receiver, read emails, check for HTTP or UDP messages, Bluetooth, and the list goes on). All of this makes interfacing with a Pi quite simple. In the direction of actuating (rather than detecting), the Pi is also capable of pulse-width modulation (PWM), which is a powerful technique for modulating digital signals. Closely linked to the idea of a 'duty cycle', PWM allows you to vary, in an analogue fashion, how much time the signal is in a high-intensity state; this in turn lets you dim LEDs, control the direction of a servo motor, and so on.

Tip 10: You can do computational biology (and other science) with it, both teaching and research

Phenomenally, given its affordability and simplicity, the Pi can be used for 'real' scientific pursuits, on both the research and teaching fronts. For example, a Pi has found its way into synthetic organic chemistry, where it has been the key hardware component in an integrated, open-source platform for the automated, machine-assisted synthesis of target compounds [40]. In that context, a Pi was used for real-time (online) data acquisition, analysis, and feedback; by running control programs to orchestrate the activities of spectrometers, reactor units, and so on, the Pi was the key module in a feedback loop for simultaneous control of multiple flow-chemistry devices. Somewhat similar in spirit, a Pi Zero was recently used to control piezoelectric pumps in a 'biocompatible, low-cost programmable dynamic flow pumping system'; dubbed 'PiFlow', this Pi-based system enables better mimicry of microphysiological environments [41]. Moving towards bioinformatics, 'PhyloPi' is a recent, 'affordable phylogenetic

pipeline' that runs on a Pi and is intended for use in HIV drug-resistance testing facilities; the hardware profile and software engineering considerations that factored into this pipeline's design, including the Pi's limited memory (1 GB random-access memory [RAM]) and quad-core capabilities (enabling parallelised code to be run), are well described in [42]. As shown in Fig 1, one can also generate publication-quality molecular illustrations using a Pi and open-source code such as PyMOL [43] for less than \$50 USD and about an hour of work, from start to finish.

On the teaching and pedagogy front, the Pi has been the centrepiece of significant efforts by Scottish research groups, based at Edinburgh and St Andrews, to develop and disseminate bioinformatics educational materials. These materials are freely distributed as an open-source project called '4273π' [44], which is also the name of the Raspbian-based OS image developed as part of the project. Focused on bioinformatics and targeted at upper-level (fourth-year) biology undergraduates, the 4273π resource [45] provides one with training in basic Linux system and network administration, the Unix command line interface, and so on—all invaluable skills in any computational science, biology, and beyond. In the same spirit, that research team has pioneered usage of the Pi as a way to teach the basics of computational sciences (quantitative and algorithmic thinking, programming principles [46]) to broader audiences, including high-school students [47] and other members of the general public [48]. Finally, as an illustration of its general utility and versatility, note that the Pi is also being adopted for instructional purposes in other areas, such as in radiology training programs [49].

Conclusion

In summary, Raspberry Pi's are affordable, they are fun to use, and they can be put to very serious uses too. The skills and principles that you learn in getting them to work will serve you in good stead in today's world. You don't have to be (or become) a wizard at electronics, programming Python, or even soldering but rather just focus on gaining a practical understanding of what's going on; then, you can muddle through to make something useful. As in much of 'maker' and DIY culture [8], cosmetics is secondary to functionality; so, though most Pi projects might be in plastic sandwich boxes and held together with adhesive tape, they work, they're reliable, and their operation is transparent (no black boxes).

The world is changing rapidly, and you can only benefit from an understanding of the basics of technologies from the recent past, the present, and the imminent future. These technologies, current and future, include the IoT, the principles behind autonomous and self-driving cars, many forms of pattern recognition (e.g. facial recognition), and indeed, much of artificial intelligence—all of these will be part of the next generation's lives (and at least the tail-end of your own). Most remarkably and promisingly, all of this new knowledge is accessible for the price of a single-family trip to the movies and some Googling. Indeed, one could argue that the educational role—and further potential—of the Raspberry Pi in democratizing and demystifying computing and technology is exceptionally valuable. The Pi's role in enabling primary scientific research is still at an early stage and is taking shape.

To end, let us quote Francis Bacon, 1st Viscount St Alban, who around 1590 suggested 'scientia potestas est' (knowledge is power). In the case of the Pi, this is roughly 5 volts at 2 amps, so it's 10 watts. Frank, you were right on the money.

Acknowledgments

The authors thank Daniel Mietchen, Lane Rasberry, Paul Rogers, John Royle, Deepak Winston, and Professor Philip Bourne for review and comments.

References

1. MagPi, The Official Raspberry Pi Magazine: Sales soar and Raspberry Pi beats Commodore 64, becoming world's third best-selling computer; 2017. Available from: <https://www.raspberrypi.org/magpi/raspberry-pi-sales>. [cited 18 April 2019].
2. Top-12 Raspberry Pi Alternatives; 2018. Available from: <https://www.zdnet.com/pictures/top-12-raspberry-pi-alternatives-2018-edition>. [cited 18 April 2019].
3. Raspberry Pi—Teach, Learn, and Make with Raspberry Pi; 2018. Available from: <https://www.raspberrypi.org>. [cited 23 September 2018].
4. Eben Upton: Raspberry Pi; 2018. Available from: <https://www.computer.org/csdl/mags/co/2013/10/mco2013100014.html>. [cited 24 September 2018].
5. ARM loses IP market share to fast-growing followers; 2019. Available from: <http://www.eenewsanalog.com/news/arm-loses-ip-market-share-fast-growing-followers>. [cited 06 February 2019].
6. iPhone 5—Geekbench Browser; 2018. Available from: <https://browser.geekbench.com/v4/cpu/11500380>. [cited 18 April 2019].
7. Raspberry Pi 3—Geekbench Browser; 2018. Available from: <https://browser.geekbench.com/v4/cpu/11296737>. [cited 18 April 2019].
8. NOOBS installation manager; 2018. Available from: <https://www.raspberrypi.org/documentation/installation/noobs.md>. [cited 25 September 2018].
9. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, et al. (2000) The Protein Data Bank. *Nucleic acids research* 28: 235–242. PMID: [10592235](https://pubmed.ncbi.nlm.nih.gov/10592235/)
10. Mura C, McCrimmon CM, Vertrees J, Sawaya MR (2010) An introduction to biomolecular graphics. *PLoS Computational Biology* 6.
11. MagPi, The Official Raspberry Pi Magazine: Breadboard tutorial: Learn electronics with Raspberry Pi; 2018. Available from: <https://www.raspberrypi.org/magpi/breadboard-tutorial>. [cited 18 April 2019].
12. Schild TD, Victor C (2018) Troubleshooting and Analyzing Network Booting Challenges Using a Raspberry Pi Testbed. In: HPC Mini-Showcase. [Accessed 8/6/2018] Available at <https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-18-27425>.
13. Johnston SJ, Basford PJ, Perkins CS, Herry H, Tso FP, et al. (2018) Commodity single board computer clusters and their applications. *Future Gener Comput Syst* 89: 201–212.
14. Di Justo P, Blakeslee J. Raspberry Pi or Arduino? One Simple Rule to Choose the Right Board; 2015. Available from: <https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/>. [cited 25 September 2018].
15. Parrish K. Your DIY project needs a computer. Should you choose Raspberry Pi or Arduino?; 2017. Available from: <https://www.digitaltrends.com/computing/arduino-vs-raspberry-pi/>. [cited 25 September 2018].
16. Raspberry Pi: Remote Access via VNC; 2018. Available from: <http://www.raspberrypi.org/documentation/remote-access/vnc>. [cited 18 April 2019].
17. Mohamed I, Dutta P (2015) The Age of DIY and Dawn of the Maker Movement. *GetMobile: Mobile Comp and Comm* 18: 41–43.
18. Tanenbaum JG, Williams AM, Desjardins A, Tanenbaum K (2013) Democratizing Technology: Pleasure, Utility and Expressiveness in DIY and Maker Practice. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*: ACM. pp. 2603–2612.
19. App Inventor; 2018. Available from: <http://appinventor.mit.edu/explore/about-us.html>. [cited 18 April 2019].
20. [Thunkable.com](https://thunkable.com): Drag and Drop Mobile App Builder for iOS and Android; 2019. Available from: <https://thunkable.com>. [cited 18 April 2019].
21. Morin A, Urban J, Sliz P (2012) A quick guide to software licensing for the scientist-programmer. *PLoS Computational Biology* 8: e1002598. <https://doi.org/10.1371/journal.pcbi.1002598> PMID: [22844236](https://pubmed.ncbi.nlm.nih.gov/22844236/)
22. Barnes N (2010) Publish your computer code: it is good enough. *Nature* 467: 753. <https://doi.org/10.1038/467753a> PMID: [20944687](https://pubmed.ncbi.nlm.nih.gov/20944687/)
23. Internet-in-a-Box; 2018. Available from: <https://en.wikipedia.org/wiki/Internet-in-a-Box>. [cited 06 February 2019].
24. Ai Y, Peng M, Zhang K (2018) Edge computing technologies for Internet of Things: a primer. *Digital Communications and Networks* 4: 77–86.
25. Open Source Initiative; 2019. Available from: <https://opensource.org/>. [cited 18 April 2019].

26. Lakhani KR, Wolf RG (2003) Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. <https://doi.org/10.2139/ssrn.443040>
27. Raspberry Pi Forums; 2018. Available from: <https://www.raspberrypi.org/forums/>. [cited 18 April 2019].
28. Catalog of All Stack Exchange Sites; 2018. Available from: <http://stackexchange.com/sites>. [cited 18 April 2019].
29. Raspberry Pi Stack Exchange; 2018. Available from: <http://raspberrypi.stackexchange.com>. [cited 18 April 2019].
30. Chemistry Stack Exchange: Molecular visualization software for Raspberry Pi?; 2018. Available from: <https://chemistry.stackexchange.com/questions/88397/molecular-visualization-software-for-raspberrypi>. [cited 18 April 2019].
31. Parnell LD, Lindenbaum P, Shameer K, Dall'Olio GM, Swan DC, et al. (2011) BioStar: an online question & answer resource for the bioinformatics community. *PLoS Comput Biol* 7: e1002216. <https://doi.org/10.1371/journal.pcbi.1002216> PMID: 22046109
32. Blischak JD, Davenport ER, Wilson G (2016) A Quick Introduction to Version Control with Git and GitHub. *PLoS Comput Biol* 12: e1004668. <https://doi.org/10.1371/journal.pcbi.1004668> PMID: 26785377
33. Raspberry Pi—Quora; 2019. Available from: <https://www.quora.com/topic/Raspberry-Pi-1>. [cited 18 April 2019].
34. 50 of the most important Raspberry Pi Sensors and Components; 2018. Available from: <https://tutorials-raspberrypi.com/raspberry-pi-sensors-overview-50-important-components/>. [cited 25 September 2018].
35. Upton L. The Raspberry Pi in scientific research; 2013. Available from: <https://www.raspberrypi.org/blog/the-raspberry-pi-in-scientific-research/>. [cited 25 September 2018].
36. Hartley T. AirPi: Air Quality & Weather Project; 2018. Available from: <http://airpi.es/>. [cited 25 September 2019].
37. Fitzpatrick DE, Battilocchio C, Ley SV (2016) A Novel Internet-Based Reaction Monitoring, Control and Autonomous Self-Optimization Platform for Chemical Synthesis. *Org Process Res Dev* 20: 386–394.
38. Public Lab contributors. Public Lab: a DIY environmental science community; 2018. Available from: <https://publiclab.org/>. [cited 25 September 2018].
39. Hall effect—Wikipedia; 2018. Available from: https://en.wikipedia.org/wiki/Hall_effect. [cited 18 April 2019].
40. Ingham RJ, Battilocchio C, Hawkins JM, Ley SV (2014) Integration of enabling methods for the automated flow preparation of piperazine-2-carboxamide. *Beilstein J Org Chem* 10: 641–652. <https://doi.org/10.3762/bjoc.10.56> PMID: 24778715
41. Kassis T, Perez PM, Yang CJW, Soenksen LR, Trumper DL, et al. (2018) PiFlow: A biocompatible low-cost programmable dynamic flow pumping system utilizing a Raspberry Pi Zero and commercial piezo-electric pumps. *HardwareX* 4: e00034.
42. Bester PA, De Vries A, Riekert SJ, Steegen K, van Zyl G, et al. (2018) PhyloPi: an affordable, purpose built phylogenetic pipeline for the HIV drug resistance testing facility. *bioRxiv*.
43. The PyMOL Molecular Graphics System; 2018. Available from: <https://pymol.org/2/#opensource>. [cited 25 September 2018].
44. Barker D, Ferrier DE, Holland PW, Mitchell JB, Plaisier H, et al. (2013) 4273π: bioinformatics education on low cost ARM hardware. *BMC Bioinformatics* 14: 243. <https://doi.org/10.1186/1471-2105-14-243> PMID: 23937194
45. 4273π: Bioinformatics for Biologists; 2018. Available from: <http://4273pi.org>. [cited 24 September 2018].
46. Ekmekci B, McAnany CE, Mura C (2016) An Introduction to Programming for Bioscientists: A Python-Based Primer. *PLoS Comput Biol* 12: e1004867. <https://doi.org/10.1371/journal.pcbi.1004867> PMID: 27271528
47. Barker D, Alderson RG, McDonagh JL, Plaisier H, Comrie MM, et al. (2015) University-level practical activities in bioinformatics benefit voluntary groups of pupils in the last 2 years of school. *International Journal of STEM Education* 2: 17.
48. McDonagh JL, Barker D, Alderson RG (2016) Bringing computational science to the public. *Springerplus* 5: 259.
49. Pereira A, Atri M, Rogalla P, Huynh T, O'Malley ME (2015) Assessment of feasibility of running RSNA's MIRC on a Raspberry Pi: a cost-effective solution for teaching files in radiology. *Int J Comput Assist Radiol Surg* 10: 1793–1801. <https://doi.org/10.1007/s11548-014-1143-4> PMID: 25547256