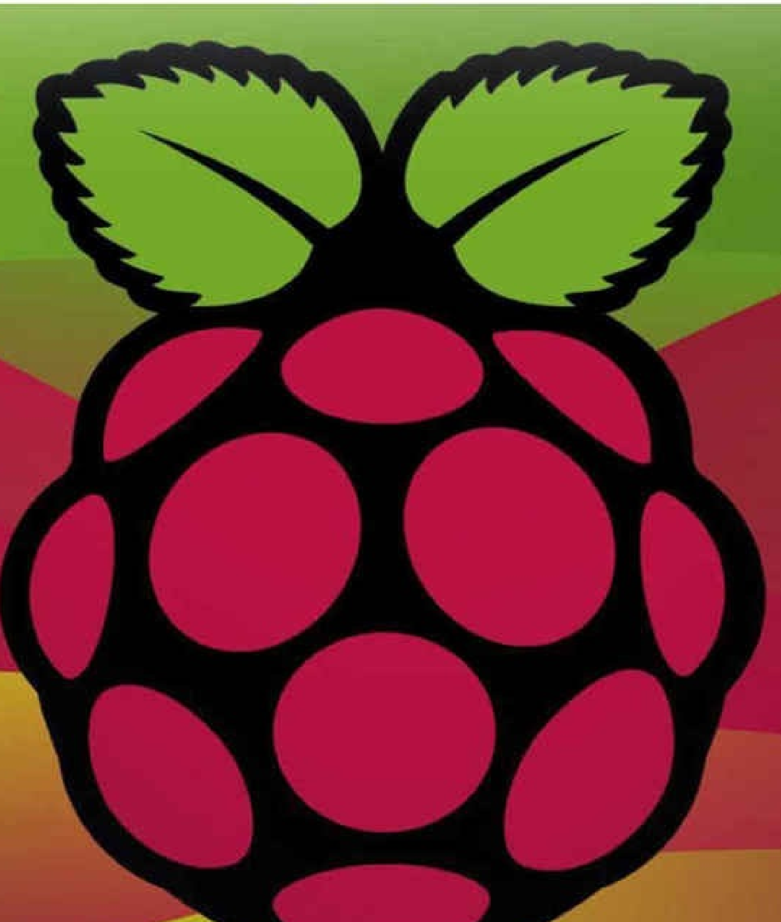
Raspberry Pi

40 Outstanding Raspberry Pi Tips and Tricks for Absolute Beginners



Dylan Day

Raspberry Pi

40 Outstanding Raspberry Pi Tips and Tricks for Absolute Beginners

Table of Contents

Copyright Notice

This document is geared towards providing exact and reliable information in regards to the topic and issue covered. The publication is sold with the idea that the publisher is not required to render accounting, officially permitted, or otherwise, qualified services. If advice is necessary, legal or professional, a practiced individual in the profession should be ordered.

- From a Declaration of Principles which was accepted and approved equally by a Committee of the American Bar Association and a Committee of Publishers and Associations.

In no way is it legal to reproduce, duplicate, or transmit any part of this document in either electronic means or in printed format. Recording of this publication is strictly prohibited and any storage of this document is not allowed unless with written permission from the publisher. All rights reserved.

The information provided herein is stated to be truthful and consistent, in that any liability, in terms of inattention or otherwise, by any usage or abuse of any policies, processes, or directions contained within is the solitary and utter responsibility of the recipient reader. Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

The information herein is offered for informational purposes solely, and is universal as so. The presentation of the information is without contract or any type of guarantee assurance.

The trademarks that are used are without any consent, and the publication of the trademark is without permission or backing by the trademark owner. All trademarks and brands within this book are for clarifying purposes only and are the owned by the owners themselves, not affiliated with this document.

Disclaimer While all attempts have been made to verify the information provided in this book, the author does not assume any responsibility for errors, omissions, or contrary interpretations of the subject matter contained within. **The information provided in this book is for educational and entertainment purposes only. The reader is responsible for his or her own actions and the author does not accept any responsibilities for any liabilities or damages, real or perceived, resulting from the use of this information.**

Introduction

Embedded computing has existed for decades. But modern developments have made the process of deploying a system easier than ever before. Thanks to the wonder of Raspberry Pi and the simple and open sourced Linux platforms that will run on it, anyone can own and manage their own server to control or automate many different basic functions around the home or in the office.

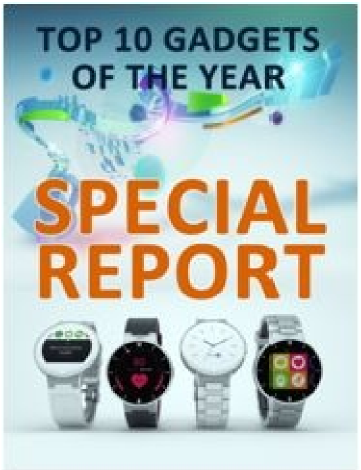
Raspberry Pi is the clear leader in embedded computing platforms. With this small, easy to program computer, you can create countless projects. Our goal in this book is to introduce you to embedded computing, explain how to program and get you started.

But first a quick word about why people should know how to code. Software development has been the path to creative disruption for decades now. One of the skills necessary is fluency in simple object-oriented programming languages. These languages empower programmers to realize the vision of new technological advances. With costs of embedded computing platforms dropping, your projects can be easy to build, inexpensive to manufacture, and profitable to sell.

So let's get started with a quick history of embedded computing. We'll tour past incarnations of small programmable personal computers that fit inside and powered other devices. We'll discuss the necessary elements of building a system. Afterwards we will go through the basics of programming in Python, a simple to learn, yet remarkably powerful programming language. Many languages are either object oriented or command/function based. Python can be either. Besides, Python also has an extensive standard library of functions that you can draw on to create your application.

Finally, I will walk you through a few simple Python-Raspberry Pi projects to get you thinking. I want to inspire you to create applications that are interesting, functional and profitable. Developing a useful tool can be the basis of a fledgling business. Maybe you can get crowdfunded to bring a product to market. After that, you could score a big investment from a venture capitalist. That investor will help you scale the production of the device and get better distribution. Suddenly, you've got a business. Soon, your business is growing and appreciating in value. Before you know it, you've got offers for your business and they aren't tiny little offers. There's a lot of zeroes in the price some bigger company is willing to pay for your idea. That's disruptive change. So, let's get disrupting!

BONUS: Your FREE Gift



Thank you for purchasing my book: "*Raspberry Pi* ". I want to show you my appreciation by offering an exclusive Special Report “*TOP 10 Gadgets Of The Year*” for FREE.

Simply Click the Button Below

Click Here to
Download the ebook

OR Go to This Page

http://bit.ly/1AiwLLM

Chapter 1 – Beginner’s Guide to Embedded Computing

Embedded computing seems like a complicated topic, so let's simplify it some. An embedded computer lives inside a larger device. It has a specific, dedicated function within that device. Embedded computers are in effect the brains of many commonly used vehicles, appliances even our homes.

A typical embedded computer consumes very little power. They are small in size and inexpensive to make. They are just a component and their specialized function is all they do. Because they can be in a variety of environments, they tend to be more ruggedly designed.

In early iterations, embedded systems limited processing resources to keep costs low and improve reliability. That is all changing thanks to innovative embedded computers like Raspberry Pi. Instead of the older, difficult to program systems that were challenging to interface with, new systems are geared toward the do it yourself and amateur developer communities.

In other words, embedded systems are now for you and me. And with them we can build amazing things.

One aspect of embedded computing that hasn't changed, is that we can still interface multiple embedded computers to existing sensors. In doing so, we better manage our resources. This allows amateur developers to substantially increase the functionality of a device with embedded systems.

Embedded systems eschew the typical computer architecture of a main board with expansion slots. Instead they rely on either a single-board computer or an embedded system on a chip architecture. The various versions of Raspberry Pi for example are based on the Modern embedded systems are often based Broadcom's system on chip microcontrollers.

What makes embedded systems most interesting is their focus. Embedded computers do a specific task. Developers optimize the computer for that function. As a result, the units can be made smaller. Additionally, these embedded computers can be mass-produced. This in turn drives the cost down. Most critically, designers build them to be more reliable than a typical PC.

Consider, if your computer crashes while you are online it's an inconvenience. If the embedded system that controls the way your car's engine works fails, your car will fail. The consequence of that type of failure could be lethal.

Consumers can find embedded computers in portable devices like their smartphones. They also drive more complex appliances like dishwashers, refrigerators or home HVAC units. Embedded computers also serve to power industrial applications. They can be as varied as healthcare equipment, automation at a manufacturing plant or more modern cars. In fact, a fairly cool project is to build a device that takes readouts from the onboard computer in a car and return a diagnosis. That way you know what exactly is going on with your car when the check engine light comes on.

And the applications don't end there. Embedded systems pop up everywhere. Applications as varied as cooking appliances, office equipment even military avionics are all powered by embedded computers.

One of the better areas for hobbyists is the realm of home automation. Built on a network of both wired and wireless devices, people can control all aspects of their home remotely or from their couch. As an added bonus, a properly programmed embedded computer system will control the most creative and inspired holiday light display your neighborhood has ever seen.

One aspect of embedded computers that make them valuable is their isolation from broader networks. Because they do not need to interact with external systems, they are more difficult to hack. Some embedded systems have begun to use wireless connectivity to communicate data in real-time. This opens them up to potential mischief from hackers. But for the most part an embedded system can be designed to function in a closed digital ecosystem. This makes them particular appealing to designers of safety and security systems as they are less likely for outsiders to compromise them.

Embedded computers can also tolerate wider temperature variations. These rugged systems are ideal for extreme environments. They are also easier to secure in water tight containers to shield them from external elements. Self-sufficient embedded computers are even able to operate when power and communication have been disabled.

One of the limited factors in embedded computers was the requirement of manually operated buttons, dials or switches to provide inputs. Think back to the old game consoles like the original NES or the Atari 2600 for an example. As microprocessors grew less expensive, they replaced analog switches in consumer products. Further advances put memory and input and output components onto a single chip with the CPU. These "microcontrollers" went into devices where a full-blown computer would cost too much.

Enter the Raspberry Pi. It's a very low-cost microcontroller that can easily handle the tasks of several components. For example, the recently announced Raspberry Pi 2 Model B includes a 900MHz Quad Core processor and a full gig of RAM. It also includes 4 USB ports, an HDMI port, a network interface controller with a standard RJ-45 port, onboard audio and video controllers and a memory card slot that works with micro SD form factor disks. It will run all ARM GNU/Linux distributions. As an added bonus it will also run Windows 10 when it is released.

The power and versatility offered by the Raspberry Pi 2 makes it ideal for developers and hobbyists. Additionally, it's $35 price tag is not prohibitive for an onboard embedded computer in a commercial product, even one aimed at consumers.

Because the Raspberry Pi manages the system hardware complexity, very few, if any components are necessary to attain full functionality. As a result, product development is largely the province of software engineers, rather than hardware manufacturers. Once a stable code base is developed, the system the costs associated with manufacturing product are fixed. As production scales, per unit costs decrease exponentially. And because software development and testing happens far faster than the development of new circuits outside the microcontroller, product design times are cut to days and weeks, instead of months and years. This gets you to market quicker than your competition.

The primary supported programming language for Raspberry Pi development is

Python. And this is not in reference to the very large snakes. Python in this instance is a hip shout out to the British comedy troupe collectively known as Monty Python. Like the Pythoneers, Python is more free form and open than its major competitor in the realm of scripting languages - Perl.

In the first decade of the 21st century, Python and Perl staged a battle for the hearts and minds of entry-level code jockeys. Neither language would ever replace C and C++ as software development languages or Java for rich internet applications. But they provided an introduction to multiparadigm programming languages. Entry level developers as well as hobbyists and do it yourself software engineers had a versatile tool to create. And create they did.

What made Python intriguing was its philosophy. Pyhton does not present an abundance of options. For example, in Python, there are extensive libraries. Those libraries provide developers an obvious path on how to do something. That led to a simpler and easier path to the goal.

Perl on the other hand allowed developers more freedom to improvise. It also set up language constraints that made it difficult to get to the end goal of development. When there are many paths, each with differing pitfalls along the way, only the most well-versed will find their way. Contrast that reality with the target audience. Perl was giving all the tools in the tool box to guys who wanted a hammer and a screwdriver. And a lot of duct tape.

Python was their roll of duct tape. Pyhton had less structure, but more obvious routes to developers objectives. Because these code jockeys were not advanced developers, that clear way forward was ideal. Perl suffered from an overabundance of versatility. Python's straightforward approach eventually allowed it to overcome Perl as the language of choice among entry-level developers.

So what is Python? It is several things. It is first widely used. It is both a general-purpose and high-level, programming language. It is also a multiparadigm programming language.

Like many languages, it was created with a foundational philosophy. Those tenets pushed concepts of easy to understand code readability. Additionally, the language allows the creation of functions and concepts in fewer lines than other languages. As a result, Python includes tools that give developers the ability to write clear programs regardless of scale.

Python's multiparadigm support includes numerous styles. Among them is the gold standard of paradigms, object-oriented programming. Most importantly for developers, Python includes a massive library of comprehensive standards.

Python includes many supports for other programming paradigms, including imperative programming. Functional programming support is somewhat limited, but still present. In addition, Python can support aspect-oriented programs. Developers using extensions can produce programs through logic programming and design by contract. The number of paradigms supported makes Python appealing to coders with a wide variety of expertise.

To begin programming in Python, you must download and install the latest release from the Python Software Foundation. After you have done so, open up the python shell, which is called the IDLE. The IDLE is where most programming in Python is done. IDLE is an acronym for the integrated development environment. When you use the IDLE, you are writing Python code.

So how about we start with the simplest and easiest to understand coding done in Python. One-line commands are programs. They use existing functions within Python to perform a simple task. Math functions already exist in Python so typing 1 + 1 at the command prompt line will return 2. Programming in Python begins with those simple foundational commands.

One of the most useful commands in Python is print. Programmers familiar with other introductory programming languages like Turing or Pascal will remember print commands. This is the same concept. Tell Python what you want it to print and it will display those words or numbers or results on the screen. So simply typing “print (“Good morning”)” will return the words Good morning in the IDLE.

So what happens when you start to combine print functions with math functions? That’s when the fun starts in Python. So let’s try this. Type the following line into the IDLE at the command prompt:

```
print (1 + 1, "can be as sad as 1. It's the loneliest number since the number 1")
```

That command will return this:

```
2 can be as sad as 1. It's the loneliest number since the number 1
```

You are now programming in Python.

Before we move onto the fundamentals of programming in Python, I want to stress to you the importance of good coding etiquette. Many software developers ignore good coding etiquette. They think to themselves, this is my code, I understand what I am trying to do here. The issue is not so much making sure everyone understands what is being written in the code. But when you hit a wall and need help, the easiest way to make sure another developer can figure out what you want to do is through clean code and comments.

Comments are in every programming language. There lines do not get executed when the program is compiled. They exist so programmers can explain what is going on inside the code to another developer. This way it is easy for another programmer to pick up where the original author of the code left off. This is ideal for collaborative projects among several different programmers, or if you run into trouble and need to get help.

Commenting in Python is simple. All it takes is a hash mark (#) at the front of your comment in the program. So let’s grab our example from above and add a comment:

```
print (1 + 1, "can be as sad as 1. It's the loneliest number since the number 1") #3dognite
```

The return remains the same:

```
2 can be as sad as 1. It's the loneliest number since the number 1
```

The commented code does not print. Comments are good to include to document features of a program. Additionally, if a bug in the program returns a useful result, by commenting and documenting it, you can turn your bug into a feature.

Which as the saying goes is the secret of all programming effort.

Chapter 2 – Basics of Python Programming

In the last chapter, we began to understand some basic programming functions in Python. I want to get more advanced, but also want to keep this at a very high-level. Python programming is best learned by doing. As you work through this chapter, spend time practicing in the IDLE, so you can improve your syntax.

This chapter is geared to enable you to learn some of the most important functions in Python. It is also going to start you down the road of programming. There are many ways to write code. Some books claim to teach the hard way, the simple way, the fast way or the right way. The reality is that there is no simple way. There is not hard way. There is no fast way. We learn coding like we do foreign languages, by doing. That's the only way to become a programmers.

As a result, I will give you the basics, explain different functions and show you how to use them. But you will only learn them by practicing and experimenting in the IDLE. So with that out of the way, I want to look at some of the most important functions in Python.

We will begin with mathematical functions. These are fairly simple. We know them from our interaction with basic math. One of the most remarkable benefits of computing power is that it has made complex math simpler. As the saying goes, Algebra was created when basic math got too big. With computers doing the math, basic math never gets too big.

So the basic mathematical functions are addition, subtraction, multiplication and division. Each of these functions exist within python already. And in the familiar forms that we know from any mathematical expression typed into a computer.

Addition is the + sign. Subtraction is the - sign. Multiplication is the _sign. Division is the / sign. Division will produce remainders, so there is a remainder function as well. It is the % sign. Finally, there is a an exponent function that will raise a number to a power. For example, 10 to the third power is 1,000 and is expressed as 10_*3.

Math functions can be influenced by order of operations. The process moves left to right across the line of code. But as with basic math, exponents are the first order of operations, followed by multiplication and division and then finally addition and subtraction.

So typing 4 + 6 _4_ * 2 returns 100 while (4 + 6) _4_ * 2 returns 160. Take a moment and tinker with these functions in the IDLE. When you feel comfortable with the mathematical functions we will pick up with loops.

In any programming language, loops are where things get done. Loops repeat certain commands until a result is achieved. One of the basic loops counts a variable up or down. The loop continues as long as a certain condition is met. This is called a while loop.

So let's practice with while loops. Inside the IDLE, let's open a new file so we can enter multiple command lines into one single runtime file. Under the file menu select new file and then type the following into the screen that appears.

```
x = 1
```

```
while x < 10:
    print (x)
    x = x + 1
    print ("We added 1 more to x. It now equals"), x
print ("And we're done here.")
```

Save the file as loopexample1 and then hit the F5 key to run the program. This will have counted the variable X up from one all the way to nine. As long as x is less than 10, the loop will continue. Once x is equal to 10, the loop stops and the next command line is run.

If you noticed, as you typed the commands into the compiler, Python automatically formatted the language for you. This is one of the things that makes Python particularly user friendly.

Next, I want to teach you about conditional statements. Conditional statements only run if certain requirements are met. These requirements are evaluated through the use of Boolean expressions. They return a true or false value for certain arguments. For example, in the while loop above, we have a Boolean expression. While x < 10 resolves as true until x equals 10. Then the loop stops. But what if we want to enforce a condition within the loop, without stopping the loop. That's where we nest the conditional statement within the loop itself. So, for example, if I want to display only odd numbers that are less than 50 and greater than 25, here's the code I enter in my runtime file.

```
Print ("Here is a list of odd numbers between 25 and 50:")

x = 25

while x <= 50:

    if x % 2 == 1:

        print (x)

    x = x + 1

print ("And that is the full list.")
```

Save the file and then hit F5 again and you have now created a conditional loop.

Within our conditional statement, there was a natural default response if the Boolean expression evaluated false. False meant don't do this. But what if we want to perform a specific command or return a certain value when the result is false? At that point we need to turn to if's partner in crime -- else.

If...then...else statements are the basics of programming. Many years ago, out of respect to the visual nature of program flows, developers began creating diagrams of the conditions within their programs. These diagrams were called flow charts. They represented a variety of potential outcomes and ensured that the code produced was nice and clean and did not lead to an unsupported end.

To clean up the code, they used if...then...else statements to provide a path for false returns in Boolean expressions. Here's an easy to understand example of an else response. Let's start a new runtime file. Type the following command language into the file.

```
a = 5

b = 12

if a == b:

    print ("a is the same as b")

else:

    print ("a is not equal to b")
```

Let's save this file as ifthenelse1 and then hit F5 to run it. The result printed on the screen is the else command. Likewise, we can continue to evaluate the expression by using the elif command, which is an abbreviation for else, if. There is no then if abbreviation, if your initial statement evaluates to true, you can nest as many additional if statements below it. Likewise, you can embed as many elif statements to create numerous branches on the decision tree. Let's try one out. We'll stick with out basic if...then..else above and make one modification.

```
a = 5

b = 12

if a == b:

    print ("a is the same as b")

elif a < b:

    print ("a is less than b")

else:

    print ("a is greater than b")
```

By adding the elif statement, we are able to determine the how a relates to b with greater specificity. The use of if...then...else statements allows for greater complexity in your programming. Through the use of loops, you can perform tasks in your program to change certain values.

Take some time before we move on to practice with loops and conditions. Nest multiple options within the program and see what happens as you go through the loops.

At this point, you are no doubt seeing ways you can produce certain results. But what is missing is user input. With information a user supplies, you can begin to generate results that when exported through a digital input/output chip on the Raspberry Pi device will allow you to control certain movements. Now you can create the functional tools for a human-machine interface. And now things start to get really exciting.

Being able to accept user input is the first step in building programs that tap into the functionality of a Raspberry Pi device. To make that input do things, we will need to utilize functions. Python is flexible in that functions can be created on the fly within programs. As a result, you can configure your program to do just what you want it to do by accepting information from an external source. Again, Raspberry Pi is perfect because it has a digital input/output to take incoming digital signals and then process the data to produce a new outcome.

Additionally, Python comes with functions built into the language. As a result, we can reference those functions in our programs. We have been making use of these pre-programmed functions already. When we reference the mathematical or Boolean operators, we making use of Python's native functions.

So let's get started with understanding functions. Functions do specific work within a program. If it is a one off function,t here is no need to define it. It's simple enough to just code the work in directly. But if it is a repetitive task the way to save the hassle of repeating a few lines of code over and over is to define the function and then reference it as needed.

Likewise, you can call any function that exists in Python to perform that task. Built-in functions are the basis of what makes Python easy to use, powerful and so popular. If I want to compare numbers, I don't need to write a function that shows the absolute value of their difference, I can just call the abs() function.

The most common built-in function is print(). But there are many others. Among them are max() and min(). These functions allow you to take inputs and return results. Since we want to begin to acquire input from user, let's start with the input() function.

Open a new runtime file in the IDLE and type the following in.

```
a = input("What is your name?:")

b = input("What is your quest?:")

print ("Hello,", a,". If you wish to", b,"surely you know")
```

c = input("what the airspeed velocity of an unladen swallow is?:") This short program will ask my name and quest and then provide a cheeky reference to Monty Python and the Holy Grail. If the inputs are based on user responses, you will need some manner of input device, keyboard, touchscreen, buttons, what have you.

Our first tip when working with Raspberry Pi is that you can run basic games on a small interface. We’ll spend more time with game sin subsequent chapters. But let’s get started with one of the simplest programs you can write to run off a Raspberry Pi.

1. Script a Mad Lib Generator.

This program will allow a user to perform string words together in Python. Because all these functions are pre-defined, you can call to each as you go. So let's write a mad lib generator.

Open a new file in the IDLE and type in the following code: #variables

```
#cycle tells the program to continue running. The while loop will run as long as
cycle equals 1.

#option is the function the user selects cycle = 1

command = 0



while cycle == 1:

    print ("Want to play a game?") print ("I promise that it's fun") print (" ")

    print ("Just give me a list of words") print ("and soon it will be done") print ("
")



        command = input("Press 1 to play or 2 to quit and then hit return:") if
```

command == "1":

noun1 = input("I need a noun: ") verb1 = input("and now a verb: ") adjective1 = input("how about an adjective: ") adverb1 = input("cool, not give me an adverb: ") name1 = input("excellent, just one more, a person's name: ") print ("Okay here's what we wrote:") print (" ")

print ("I went walking along a ", noun1, "one afternoon. My pet dog", verb1, "with me through the", adjective1, "afternoon. It was a", adverb1,"hot day and I missed my friend", name1, "very much.") elif command == "2":

cycle = 0

print ("Well that was rather silly, wasn't it? Bye bye!") Save the file as madlib and then hit F5 to run. You can make up your own silly little mad libs as you go. And even set the cycle to different numbers with each iteration of the program so your user is constantly entertained by the silly stories the mad lib generator spits out.

Chapter 3 – Practical Pi Part One

Raspberry Pi allows you to accomplish tasks around your home that you would ordinarily need other equipment to do. But rather than go with the commercially available, off the shelf technology that most folks buy, Raspberry Pi let you build a unique system entirely of your own making. What’s also nice is that these tools when constructed will allow an enterprising person to construct inexpensive platforms that can be mass produced for others. Now, instead of a do-it-yourself hobby, we’re talking about a possible business. So to begin our deep dive into the realm of Raspberry Pi tips and tricks, we’re going to explore several practical applications for the amazing embeddable microcomputer. Unless otherwise noted these projects are designed to run in the Raspbian operating system.

Before we get started on the projects, I want to break down many of the commands we’ll be using in the terminal shell of Raspbian. Raspbian is a freely available operating system based on the Debian. What makes Raspbian different from Debian is that it has been optimized to run on Raspberry Pi hardware. Like Debian, Raspbian comes with several thousand programs pre-installed to maximize the utility of your computer hardware. Linux builds typically include preset packages that will allow for users to customize the functionality of their computers. Raspbian includes admin utilities, communication packages, games, even a multimedia player.

But before we can utilize these programs, we need to set them up. And to do that, we need to use a common command prompt: sudo. Sudo stands for “substitute user do” and in effect allows a user to perform commands as the system administrator. When combined with the application packaging tool, users can set up packages in Raspbian (or any Linux OS for that matter).

The command line package management tool in both Debian and Raspbian is called “apt-get”. A user can also update, upgrade and install new packages. A typical command sequence will be sudo apt-get update for example. While I plan to share python code and command lines, this is not intended to be a book exclusively devoted to programming. So for the forthcoming projects we will go into an explanation of the hardware setup and a structure of steps to setup the Pi to do what we need it to. These projects have been curated from various sources online. Detailed, step by step coding and command instructions can be found from those sources.

1. Build a Print Server for your Home Network

Laptops wirelessly connect to your WiFi router. And with battery life better than ever, they are ideal for running without any capable. Until you need to print something out. Then you need to scurry into the room where your printer sits, hook the laptop up and then print. And like that your wireless computing is done. Not to mention, your smartphone and tablet don’t have USB ports to print from. And unless you want to get a kludgey tablet that does have a USB port, you will need some kind of print server that all your devices can recognize. Enter your Raspberry Pi.

Raspberry Pi is built to run Linux. Like other Unix based operating systems, Linux will run a program called CUPS. CUPS used to stand for Common Unix Printing System. CUPS was the print engine of choice for Linux boxes and remains so now. When Apple turned its OS into a Unix derived system in 1999, they began to utilize CUPS as well. Apple later acquired the source code to CUPS along with the services of CUPS’ chief developer. As such CUPS is native to Macs, Linux boxes and works well with iOS devices. That makes CUPS perfect for a print server on your Pi.

The first step is to install CUPS on your Raspberry Pi. Make sure your Pi is connected to the Internet and download and install CUPS. After completing the download and install process, we will need to give the Pi user access to the printer and printer queue. Finally, we need to allow remote access to the CUPS server. To begin we use the sudo command line with apt-get to install CUPS. Once CUPS is installed, we again use sudo to modify the lpadmin utility that configures CUPS. Finally, we use the nano text editor to edit the instructions file for CUPS where we will need to replace the “Listen localhost:631” with “Port

631”. Now CUPS will pick up any contact that is sent to Port 631. Finally in the locations section, you will need to add “Allow @local” to the following locations sections:

```
# Restrict access to the server

# Restrict access to the admin pages

# Restrict access to the configuration files
```

Exit from the instructions file and save the changes. Now we reboot the CUPS server. The command line you need to use to accomplish this is: sudo service cups restart

After CUPS restarts, you can login to CUPS from any local PC using any browser and entering your PI’s IP address with “:631” following. The admin panel is fairly self explanatory. Make adjustments and run the required driver and you have a print server. Test this from other PCs on your local network and now you can freely print from anywhere.

2. Setup a home media server with XBMC

This is one of the easiest and most popular Pi projects you'll encounter. It requires almost no coding. However, we will need some additional hardware. I am going to assume you have a Pi, a power supply with at least 700mA at 5V, a case if you want one, a wired keyboard and mouse and cabling such as CAT-5 for Ethernet and HDMI for video. To that we will need to add a USB hard drive (I prefer having the media attached to the Pi, plenty of folks who have done this build just stream from network connected PCs), an ultra high speed SD card with at least 16GB of storage space, a remote control and an installer for raspbmc. You can use other versions of XBMC, but raspbmc has given me the best results thus far.

We begin by loading the installer onto the SD card. Once you have raspbmc on the SD card, we can attach your Raspberry Pi to your television. Plug the cables and SD card into the Pi in the following order: 1) HDMI cable from the Pi to the TV

2) Ethernet cable to the router 3) SD card in the Pi 4) MicroUSB power cable into an electrical outlet One the Pi powers up, it will boot from the SD card and install raspbmc. When the installation is complete, the Pi will restart and load XBMC. Now we can adjust the settings to optimize the functionality of the media center. Start with setting the resolution for 720p. Videos at the 1080p resolution are almost always choppy. So stick just with 720p. If your install of the media center exceeds the size and form factor of the television, you can adjust the overscan settings to make it fit. From the system performance profile, go ahead and goose that up to fast. We're going to go into greater depth on overclocking your Pi later,

but for now, just rest assured that overclocking this computer is an approved use. From here you will need to add videos to your library – this is why I like having the videos stored right on the USB hard drive, makes for easier access.

Create a digital information center

Up to date information is extraordinarily useful. The nice thing about this application is that you can literally put any information that is easily accessible from a website. So if you want recent posts to Facebook, tweets from friends or the folks you follow, new articles posted to your favorite websites, or up to date digital radar for the area around your home, as long as you have installed chromium on the Pi, enabled Kiosk mode and have the URL of a page to display information.

Now the clever sorts among you are already pondering how you can pull together multiple information sources into a specific amalgamation of detail and data. Something like the way that you can create a real time news ticker for a website and a real time twitter app that shows the most recent tweets and other such consistently updated information. Maybe throw in a weather map or traffic or transit updates. Now you never need to wait for traffic and weather together on your favorite radio station. You have your information kiosk with all of that, plus headline news.

So let’s get started to set up the Pi to do our bidding and help us organize the important information we need in the morning to get out the door. As with all these projects it begins with a Pi. To it we add some hardware:

1 HDMI cable – if the display you want to use isn’t HDMI, that’s fine, just make the cable an HDMI to DVI or VGA cable

1 USB WiFi adapter

1 MicroUSB charger

1 Monitor

At this point we are ready to boot up to the setup screen. We will want to make some changes. If this is a new Pi, you will need to update the locale, timezone and boot behaviour (and yes it is spelled that way) to that it boots to desktop. You can also change the password, but you do not need to. You should also expand the partition on the SD card to maximum. Finally, we will want to update the config file and reboot the Pi.

When we reboot, the Pi will request we log in. If you did not change the password in the setups above, then your login should still be “pi” and the password should be “raspberry”. Next we need to update the Pi and install Chromium as our browser. Just as we did with CUPS, above, we use sudo in conjunction with apt-get to update and upgrade and then install chromium.

Your Raspberry Pi will take some time to do all of this. You can sit waiting or find something productive to do. Once it is all done, we will want to name this Pi something appropriate. We will do this in the config files, but we can quickly change the hostname by using sudo hostname *newhostname* where we replace the italicized newhostname with the new hostname we would like to use. How about we use infokiosk as the hostname? Sounds good to me too. Now we want to update the config file to load chromium in kiosk mode upon restart and to suppress the screensaver. We can do this in nano. We can also do this in pico. I prefer nano, since it is licenses as part of the GNU general public license. So pull up nano with the autostart file.

Then from there you will use a hashtag to turn the command line for a screensaver into a comment. From there we want to set chromium to load up with the correct page. Use the kiosk command in the file to make sure it is pointing to the webpage with the information we want to show in the kiosk.

If only we had a computer to host and serve up a webpage with this kind of information. You know with a news crawl, traffic and weather detail. It wouldn’t need to be a powerful web server, just a simple pass through of data from other sites. Oh wait. Yeah we have just that kind of computer. We just need to…

3. Turn your Raspberry Pi into a web server

Building your own webserver can be daunting. And Raspberry Pi is not up for major hosting requirements. If you want to build a site to draw in lots of traffic, you need something with more power than the Pi can offer. However, for our kiosk idea, setting up a web server on the Pi is an easy and effective means of creating the right page details for your kiosk. No one else will be accessing the website unless they happen upon the IP address you create. You do not even need to request a domain name, though as I will explain in a minute, this step is a good one to take.

Here is the basic list of things we need for this build: I suggest a UHS class 1 SD card, ideally 16GB, but 8GB will work. Connect to your router through Cat-5 cable. Also, port forward one of the ports from the router from the external IP address to the Raspberry Pi on that port. It’s time to enable the root user. That command is easy: sudo passwd root *password* No, don’t make it password. After you have created the root user, we will need to perform the same series of updates and upgrades. From here we get to start installing Apache, PHP and MySQL to set up the web server and database. We’re going to need to use our apt-get command prompts to install these. And bear in mind this will take quite some time to unpack and install each program.

After MySQL is installed it is time to setup the database. As MySQL was installing you were asked to create a password. That should replace the italicized password below: mysql -uroot –p*password* Now we get to create the database itself.

mysql> create database *dbname*; At this point you are able to setup the space for hosting the website. Here are those commands: chown -R pi:pi *var*www chmod -R 0755 *var*www And now we can install the file transfer protocol client. And we want to make sure that we are using SFTP so the transfers are secure.

apt-get install vsftpd –y When that finishes configuring, we will want to get the FTP client booted up. And once the FTP client is up, start Apache. At this point you can load Joomla or Wordpress or another web content management system to begin populating your web page.

4. Use the Pi to own your own name

Many people set up websites for ecommerce, blogs or a means of sharing information and data. One of the best reasons to run your own website is to market or promote yourself. Sometimes it’s simple. Like, for those seeking work, especially in the tech fields, this is the ideal way to make a great first impression. You can share with a prospective employer your personal Pi site and invite them to take a look.

On it, you are displaying your ability to code, manage technological applications, even impress them with a specific subsite to highlight how you could help their business. And it is far more effective than a paper resume. But of course, give them a PDF of your actual resume that they can print out from your server. What’s better is if you are doing career counseling, you can set up other folks with their own personal marketing site to promote their career prospects.

More than that though, make sure when you set up your website, you register the domain name with your own name. There is a prominent politician who didn’t do this. As a result when you type in that politician’s name into Chrome with “.com” after it, you are greeted with a webpage that promotes policies antithetical to the ones held by this politician. It’s a rather foolish to enter the public space without securing the rights to your own name on the internet.

This politician’s folly is a lesson to folks like you and me. Your Pi site is an opportunity to own your own name. And when you own your own name, you control the messages disbursed in your name. Any politician or public official who fails to own his or her own name is not worthy of your vote. Their lack of vision and forethought is enough to diminish their qualifications to be a steward of the public trust.

5. Slap a Pi on the back of a Monitor

The small form factor of the Pi makes it an ideal option to attach to the back of a standard desktop monitor. Doing so allows you to create a mock all-in-one computer that reduces cable clutter and keeps your desk clean and tidy. You can substitute parts as it makes sense. But at a minimum you will need things like the following:

- Raspberry Pi
- Enclosure
- USB Wi-Fi adapter
- LCD monitor with built-in powered USB hub, HDMI, speakers
- Wireless keyboard and mouse
- One foot long or less HDMI Cable
- One foot long or less USB A to USB B cable
- One foot long or less USB A to micro USB cable

You can also benefit from some cable ties and Velcro to keep the cabling out of sight. While the Pi will never be confused with high end Power Macs, you can enjoy the clean design and minimal clutter that Apple puts into these hyper expensive machines for less than $150 of gear from the local consumer electronics store or out on Amazon.

Here’s how we set it up. First, seat your Pi in its enclosure. Then attach the Pi to the back of the monitor with Velcro straps. Next connect the micro USB cable to the Pi and the powered USB hub on the monitor. The Pi is being powered off the monitor itself. Attach the HDMI cable to the Pi and the monitor. Connect the short USB A to USB B cable from the Pi to the monitor. Run the audio cable from the monitor to the Pi. Attach the USB dongle for the keyboard and mouse to the monitor. Attach the USB WiFi dongle to the Pi. Once everything is connected, the cables can be bundled together with cable ties and velcroed to the back of the monitor. And with that you have a do-it-yourself all-in-one PC powered by the Raspberry Pi.

6. Internet Radio on the Go and the Pi

7.

An intrepid programmer expended almost 300 lines of code to create his own Pi based internet radio receiver. And remarkably, the radio functions similar to any regular radio. With presets built into the device and push button functionality. That's great. As a radio nerd from years gone by, I am in love with this project. But I want to take this a step further. I'm somewhat crazy like that.

We can mount this rig, complete with the Pi into a housing capable of fitting into a car's dashboard. That's right. We're going internet radio in our car. We'll need a MiFi device to provide signal to the Pi. We'll also need to connect buttons on the dashboard to the inputs on the breadboard. We'll also need to patch the power through to the Pi direct from the vehicle's electrical system. Finally, we'll need to make sure the WiFi adapter on the Pi syncs with your MiFi. But once that is done and tested, you have a dedicated car stereo that draws on a playlist of Internet radio stations from all over the world. If you find the terrestrial radio options lacking and don't really care for the canned radio from the satellites above, then going to the Internet is an ideal option.

As cars are equipped with WiFi with greater consistency, the added equipment of the MiFi device will become an unnecessary add-on as well.

Chapter 4 – Practical Pi Part Two

We’ve built some fairly useful Pi applications in the previous chapter. In this chapter we’re going to pump things up some and explore some projects that will allow you to branch out and explore added levels of creativity in your efforts to make the most of your Pi. We’re going to start off with what has become the holy grail of modern electronic development.

1. Setup a home automation system using your Raspberry Pi.

Home automation means a lot of things to different people. What we are going for is a system that is managed by your Raspberry Pi, ideally with a remotely accessible, web-based interface that you can pull up on your android or iOS smart device to switch on and off lights and other appliances. This won't really work with devices that require a one-time touch to start them. This is more for the dedicated rocker switch like you use for lights or other AC appliances. If it can be left in the on position, this type of a rig will work just fine.

Our hardware list includes a Raspberry Pi of course. We'll also need remote controlled outlets that we can plug our lights and appliances into. And then we'll need to load onto the Pi an application that has been built in Node.js. Node is a JavaScript interface that allows JavaScript to be run on a server rather than the client. That's a total inversion of the way JavaScript typically works. So using sudo we will need to install node in addition to updating and upgrading Raspbian. From there in the profile, we will need to update the export path to call out where node.js is located. After node is installed and configured, we will need to add Mongo. Once Mongo is in place we can finally add the home control program. The program itself was written in JavaScript to run on node by a developer named Willi Thiel. His app is freely available on github. You can enter the commands to get the program from github and install direct from the command prompt. The app is named heimcontrol.js. Once it is installed and running on the Pi, you can now attach the hardware.

Crack open the remote for the outlets. Inside you will find the radio control chip. We will want to wire this up to the Pi via a control board. Unless you have experience building printed circuit boards, your best bet is to use a pre-built controller board like the Arduino Uno. Once the RC chip is wired to the Ardruino, you will need to load up the program on the Arduino to listen for commands to come from the Pi. Those commands will be played down the line from the heimcontrol program that we will run either directly off the Pi, or by remote from a smartphone that hits the IP address for the Pi via a browser.

This build has a lot of wires on the back end – an odd occurrence, considering the outlets we are using are wireless. But that can be cleaned up by building a small enclosure and mounting the Pi and the Arduino board within the enclosure. If you use metal for your enclosure, mount the remote control chip on the exterior.

2. Indoor temperature and humidity monitoring station

I’ve found that there are only two temperatures in my home – too hot and too cold. That makes for rather uncomfortable times when out in the kitchen cooking, especially when the humidity from boiling water or long simmering soup adds to the discomfort of the already warm temperatures. From my experience, thermostats that trigger the cooling breezes of a centralized AC unit are horribly located. And it also leads to the people who aren’t in the kitchen to shiver in the parts of the house where the humidity and heat aren’t so bad.

It’s the same phenomenon that you likely have encountered on longish drives on sunny days. The person on the sun-facing side of the car broils no matter how cold the air conditioner is set. The person away from the sun needs a parka. If our cars have dual zone climate control, shouldn’t our homes? This application will not give us dual zone climate control, but what it will do is create a record of the temperature and humidity throughout the day to enable you to adjust your (inconveniently located) thermostat to correct for the fluctuations.

As with home automation, we need hardware. The ideal option is a temperature and humidity sensor with a digital interface. These are relatively inexpensive and designed typically to work with both Arduino and Raspberry Pi. Because the Pi has digital inputs and outputs on-board, the sensor can be hardwired into the Pi or plugged into a breadboard and wired to the Pi from there.

The interface is controlled through code. Python and C are good languages for getting the Pi to display the temperatures. Setting up a recurring job on a schedule, these are commonly called cron jobs, will allow you to have the Pi write the data recorded by the sensor to a file. With our log file, we can use any utility to graph data points to create a readable chart.

3. Alright, let’s take temperature sensing and turn it into home automation

It’s one thing to set up a remote control that turns lights on and off. It’s something else to collect data readings within your home. The flipside is using the readings and some of the home automation skills we developed in the first project of this chapter to wire up a fan or other air circulation device to the remote controlled outlets. Then, when the temperature in a room exceeds a certain pre-determined level, a program sends a command to that remote controlled outlet to turn the fan on. While it won’t decrease the temperature, it will improve airflow and help the occupants of that room feel more comfortable. These types of applications can serve several purposes. But for those we may want to build out our collection of data to include extra weather readings. And rather than restrict our collection to only the readings we can take inside, let’s gather some details from the great outdoors.

4. Attach a weather station to your Pi

The advantage of a weather station is that we can collect more readings than just humidity and temperature. Those additional readings make it possible for us to log more data, plot more data and use those data to trigger other actions from our home automation. In fact, by combining the data readings from our weather center with the information kiosk, we can provide up to date weather readings on our morning info board. The limits for such a project's practical applications are nearly limitless.

Let's begin as we always do with hardware needed. First, one Raspberry Pi will be required. Second, the weather station itself is necessary hardware. The first and most important aspect of any weather readings you take is that the sensors should be in digital format. If the signals are in analog, they will need to be run through an analog to digital converter. Rather than spend time specifying an analog to digital converter, we should make sure our sensors output in digital so we can feed them straight into the Pi or via a breadboard into the Pi. The weather station should include at a minimum:

- Exterior mounted sensors for temperature, humidity and barometric pressure

- A rain gauge

- A wind speed and direction gauge

With those five you can collect and log various readings related to temperature and current conditions. And when correlated to past weather events, you can spot trends. Plenty of amateurs collect and stream data related to weather to their local news stations. Hooking the Pi up to these kinds of sensors enables you to quickly and easily use the same interface, logging and plotting controls to track historical data.

First, we’re going to set up our sensors outside to take their readings. Make sure to test using any displays that were included with the weather station. Once the sensors are mounted outside, wire their receivers up to the breadboard or directly to the Pi. If you want to make things simple, download and install a weather station reading program. If you want to display the local readings with a weather forecast, set the Pi to display your sensors output along with weather forecasts. If you have your Pi connected as a webserver, those readings can be displayed on your information kiosk or on a page you access from another location via your phone or a PC.

5. Place and receive calls via the Pi

One upon a time, the biggest promise of broadband internet was computer based telephony. This was the biggest pursuit of hobbyists and businesses. They would take us all off the PBX’s run through old ma belle’s copper backbone. In its place would be packets of data that could be converted in real-time into crystal clear voice communications. The biggest benefit to subscribers would be a flexible network that automatically re-routed calls through many hubs and nodes and datacenters at lightning quick speed. Now downed transmission lines wouldn’t stop calls. But somehow with even quicker speed, voice over internet protocol applications were supplanted by cellular phones. And quietly, unbeknownst to most land line subscribers, their phone service went from copper and fiber to internet relays.

Yeah, that’s right. The vast majority of landlines are already using a VOIP interface. But that VOIP interface is controlled solely by a telecom provider who would prefer to keep you on their hardware and away from something that liberates you, like the Pi. Now plenty of competitors exist. Companies sell USB jacks that plug into a computer and let people dial out. Other companies provide you with a VOIP box that connects to your router and allow you to plug a phone into the box to make outgoing calls. Still other companies market high-end digital voice services to businesses that allow them to monitor calls, record everything using a software interface and best of all, do so on the unlimited bandwidth of their cluster of T1s.

But if you want to free yourself from the claws of the telecoms who bill you for their crappy VOIP service, your best bet is to do it yourself and build out your own Pi phone service. The process is relatively simple and requires only a few pieces of hardware and software.

First, you will need a Raspberry Pi. Second you will need either a software based SIP phone or an interface that allows you to connect a standard phone to your Raspberry Pi. The interface often works best as they allow you to use your own phone with the network. The other reason why this is ideal is because the technology is made in mass quantities because most telecoms and cable companies include such SIP devices with their standard internet, cable and phone service. Take advantage of economies of scale whenever possible.

The next thing you will need is a Google Voice account. Verify that you can dial out from the Google Voice interface, rather than a phone. Once you confirm this, remove all the phones from your Google Voice account, but leave Google Chat as a viable option. Now, make sure you turn off Google Chat in your Gmail account. Once you have done this, you will be ready to load up the necessary software on your Raspberry Pi. Begin with the proper dependencies for the utility freeswitch, which is available for download on github. Then download freeswitch. You may need to compile the software before you install. This process will take a few hours. Once freeswitch is installed, you can configure and setup freeswitch so it associates to your gtalk account using your gmail address and google password. At this point, you can restart freeswitch and test it out. If everything works, return to the shell and plug the IP address for your SIP device directly into the domain name line of the default directory xml file.

Having completed this setup, you can set up your SIP device to communicate with freeswitch on the Pi. Also, you will be able to setup outgoing callerID information. Then you can start freeswitch back up, reboot the SIP device and then add the SIP device location to the default file in the dialp directory on the Pi. Once you finalize your configuration of the dingaling utility on the Pi, you are ready to hook everything up and start making calls. If everything is working, set freeswitch to autoload on the Pi and you have your own homemade voice over internet protocol phone.

6. Snap a picture with your Pi

Most digital cameras are computers with a lens attached. We don’t think of them as computers, but they are. This is why the ultra compact Raspberry Pi is so ideal to be a camera. Nicely enough all the equipment necessary to make a Raspberry Pi digital camera is readily available. So let us begin with the hardware list.

First, (by now you have guessed this already) one Raspberry Pi. Secondly, we need a touchscreen for the Pi. Several exist. The primary project on which this write up is based calls for a PiTFT kit. You also need the Pi Camera Board. Get an SD card with at least 8GB. Youc an also set up a WiFi adapter or a USB battery pack to extend the life of the camera.

You can solder the touchscreen to the expansion header. You can also build an enclosure for the camera. These are optional but highly recommended. The camera itself will be unwieldy if you don’t have it contained somehow, even if it just some rubber bands. Once you have your hardware and your camera containment, it’s time to set up the Pi.

You will need to enable the options for the expanded filesystem and for the camera itself. You will want to set the time zone and keyboard to match your regions. You need to switch off overclocking if you have set it up previously. You might want to give your Pi camera an appropriate name. After this set up the display and make sure you are familiar with how it should work. Once you do that, you need to get and setup the Python library for the camera application. Once you boot the camera software you are ready to use the touchscreen to look through the camera lens. If everything looks right we can finalize the settings by setting up your Raspberry Pi Camera to autoboot into camera mode. Now you’re ready to take pictures. Oh, how does one take pictures without a shutter button? That’s easy. The upper three-quarters of the viewfinder screen is the shutter button. So point, snap, save and enjoy your own do it yourself camera.

7. Camera accessories, including professional quality rigs and dollies

One of the best applications for the Raspberry Pi is setting up control over physical objects. As we explored briefly in the project above for the home automation, the digital input output controls on the Raspberry Pi can trigger power to come on to various appliances. But we can do more with the Pi. Digital I/O is the heart of motion control and various robotics platforms. They allow for sensor gathering and other means of controlling the discrete motion of objects attached to a motor. The controls are commonly programmed, but you can utilize stepper motors as well.

In this instance we are going to merely discuss the hardware necessary to build a dolly that will re-position a camera. This can be programmed in advance, or it can be remotely controlled. First, let’s collect the hardware. This application requires some python programming, but let’s worry ourselves with the gear primarily and let you work out the coding. For starters we need the Pi. We will also need gear to allow us to build a professional rig. That begins with a motor, rails, a platform to mount the camera on, a timing belt and a pulley system.

We need to assemble the hardware to support the camera and allow it to move smoothly along the track. This process will ensure you get steady and unbumpy video. If you are looking to capture time lapse stills, this will work as well to give you nice and smooth progression from one point of view to another.

Chapter 5 – Practical Pi Part Three

We’ve built some really neat things. At this point, we’re going to take things into our final realm of practical purposes. And one of the most important things a practical Pi application can do is to provide something simple and easy in a way that makes it irreplaceable. Let me give you an example. I love coffee in the morning. And more on point, I need coffee in the morning. I’m always exhausted when I get up. And I frequently mess up making a common cup of Joe to get me going in the morning. But what if we could use the input output capabilities of the Pi to run our coffeemaker? Then all I would need to do to make sure I had a fresh hot cup of coffee ready for me is measure everything out the night before. That sounds like a worthy project.

1. Automate your coffee machine with Raspberry Pi

We begin with the hardware. We'll want a regular run of the mill coffee machine. Like our work with home automation, we can set those old fashioned appliances with the dedicated rocker power switch to start when we turn on power. For those with a momentary contact switch or an electronic relay to start the device, the automation we have won't do. We can rely on the home automation setup we used in the previous chapter. But with the on-board I/O controls, we can use the Pi, some relays, a coffee machine, some cabling, and a little bit of code to get coffee brewing without any hassle in the morning.

The basic code is easy enough to source from the original designer of this project. But the key is to make sure you get the right kinds of relays. Some coffee machines require additional power input. The reason they do is because the heating element required to get water hot enough to brew coffee requires significant power output. As a result, in adequate relays can potentially fry your Pi. So make sure you spend a little extra and get the good equipment.

With the ability to brew coffee covered, your mornings are bound to be better and your projects will be certain successes.

2. Monitor the status of your network by Pi

The familiar standard that we all hate is when we boot up the computer, we wait for it to recognize and then authenticate our wireless network. The delay is most of the time seconds. Occasionally it stretches out for a minute or longer. But the end result is we aren’t sure if our network is down, infrequent as that may be, or if our computer is just taking its time to get started. Wouldn’t it be useful if you could look up and see whether or not the network is working based on a visual signal that is driven by your Pi?

So that was a project that a few Python experts undertook a while back. They attached a relay board to their network and fed the connectivity signal down the line to the Raspberry Pi. Using some Python code, they used the network status to turn a light on and off that indicated connectivity.

Now you don’t need a stop light. You can use a simple lamp. Have the Pi attached to your router and use the breadboard to connect the I/O to an outlet that turns the lamp on and off depending on the current state of the network. The code is easy to track down. Instead of guessing make sure you know the status of your network even before you fire up your PC.

3. Turn your Pi into a webcam server

Setting up a USB webcam and attaching it to your Raspberry Pi is easy enough. But by taking the Pi and making it a web server, which we know how to do from a previous chapter, and then running the camera signal up onto the Internet, we can use the Pi to monitor our property, or observe wildlife in their natural habitat, or just keep tabs on who goes into the fridge to eat our goodies while we aren’t at home. For hardware, the list is simple enough. Begin with the Pi. To the Pi we need to add a wireless Ethernet adapter and a webcam. It also makes sense, if we are going to position the Pi outside to get the nature pics that we provide some kind of water resistant containment. We also need power. Preferably steady AC power from a nearby outlet. However, if we do not have AC power available, using a battery to supply steady power to the Pi and the WiFi adapter and the webcam is a must. Also, since we have the WiFi in a USB port, we are going to need a USB hub to hook up the webcam. So that’s our hardware list.

Software starts with a Linux OS. I have used Raspbian to do this, but sometimes you need a leaner OS. With its minimal boot up time and low power draw, Arch Linux is a near ideal OS to run a webcam server. Make sure the wireless network adapter you acquire can work with the OS you end up using. That is the difference between a great project and a horrible experience. Secondly you will need utilities like motion (so that the camera is triggered by motion, rather than running nonstop) and ffmpeg, which will allow you to record and stream the video the camera captures.

Now, the Pi will need to pass the files back to a Windows PC at some point. For that I recommend using PuTTY, which will set up an encrypted, secure shell connection and WinSCP for the actual file transfer. You now have a webcam server with some inexpensive components and some free software you can download and install on your Raspberry Pi and Windows computer.

4. Use you Pi to setup a network attached storage drive

Document management is the bane of many companies and people. The files need to be in a centralized location, because you're using multiple computers and everyone needs to have access. Some turn to the cloud (and we'll head to the Pi cloud soon enough). But most people just rely on regular network attached storage. The drive that is separate from everyone's PC, but accessible by all of them. Most of the time a network attached storage rig is an expensive proposition. And if you need instant access to large files or to stream video content over the network, then you should stick with the pricier systems. They are designed to work that way. But if you just need to be able to pull files on a couple of PCs and speed isn't your main concern, using a Raspberry Pi is an easy and simple way of doing this. Best of all, this particular project requires a minimal amount of coding.

For hardware, we will line up a Raspberry Pi, make sure it is at least a model B. You will also need an SD card for the Pi to boot from, a USB hard drive that you will attach to the Pi and either a LAN cable or a Wireless Ethernet adapter to connect the Pi to your network. Given the already slow speeds and low power draw of the Pi network attached storage device that we are building, let's just leave it wired.

Now to turn the Pi into a server for the files on the drive, you will need to setup a program called samba on the Pi. We'll go back to our old friend apt-get to call samba, set it up and run it. Once samba is running, you will need to configure samba clients to serve up files to one of four possible clients. They are Windows PCs, Macs, Linux PCs and Android Phones and tablets. Since iPhones and iPads are basically allergic to files being added from anything other than iTunes or via email, you will have neither the ability nor the joy of using the Pi to serve up files onto an iPhone. With that, attach all the hardware, power up the Pi, run samba and then pull up the files from the remote storage server whenever you need them.

5. Setup your own Home V-Pi-N

So you like Starbucks and all the free WiFi you can gobble up along with the coffee you are gulping down. Have you thought for just a moment that while you bop from site to site while sipping your latte that AT&T and Starbucks are watching what you’re doing? I don’t want to get all Rockwell on you (Somebody’s Watching Me) but yeah, somebody is watching you. Every move you make on publicly available, freely accessible WiFi is watched.

Now, I know what you’re saying, the NSA is watching all of us anyway, so what does it matter if AT&T and Starbucks are doing the snooping too. Well if you are doing banking or engaging in other activities where your identity may be compromised, the question of who is watching does matter. The more relevant question is why not just cover up those prying eyes with a virtual private network. And now, even if you are using hotel internet, the only thing they’re seeing is someone on the network. Are hardware list is similar to the last project, minus the hard drive. One Pi plus one SD card, plus some CAT 5 equals a full parts list.

Before getting started, you will need to make sure you have NOOBS set up and are running Raspbian. You will need your Pi to have a static IP address, so use the instructions from your router to set that up. Setup the Secure Shell so your data transfers are encrypted. Finally, all traffic that hits Port 1194 on your network needs to be redirected to the static IP address you set up for your Pi. After that you can finally get down to updating and upgrading your Pi and installing OpenVPN. There is some code associated with this project, which you can find and add to your Pi to ensure that your browsing is only your business and your business alone.

6. Take the VPN a step further and head down to the deep web

So the starting point for anonymous browsing is the VPN. But you can go further down the rabbit hole by setting up a Tor connection on your Pi. Some hacks show a way to have both a VPN and a Tor connection. In this project we are solely concerned with the Tor connection. What makes Tor so secure, and it is gradation here. VPN is encrypted, Tor is encrypted and data packets go through multiple layers before reaching their destination. Tor browsing is the slowest of all connections, because of how much further the data packets must travel before they reach their destination.

You will need a Pi, some CAT 5 cable, a WiFi adapter, an 8GB or bigger SD card, and a power supply for the Pi. Get your Pi prepped as you normally would to access the Internet. But then we are going to update and upgrade the Pi and then we are going to install Tor. Once you complete your configuration of the Pi with Tor installed, your web activity will be buried beneath all the layers of the onion relays (yeah, that’s what Tor stands for).

Chapter 6 – Practical Pi Part Four

We’ve explored some neat utilities, ways to get and display information and how to control equipment. But what we’re going to uncover in this chapter will help you see the many ways that a Raspberry Pi is ideal for building gadgets. Much like the digital camera we built with a touchscreen, we’re going to explore how to setup consumer electronics with the Pi at the heart of the gadget.

1. Make your own Pi tablet

Now this has been studied in some depth. The basics are start with the Pi, attach a touchscreen. You will want to go with a touch screen that is big enough to give you some good space to display images, apps and do work. In addition to the touchscreen, we will want to set up the Pi with a Bluetooth adapter. This can be a dongle, but remember, the tablet will be enclosed in either a plastic or metal box. So make sure the bluetooth adapter can connect transmit beyond the box. You will also find it useful to orient the USB power jack towards the edge of the device so that the tablet can charge.

Storage will be via an SD card. The SD card will also boot up the OS (Raspbian of course). We will also need a WiFi adapter, and just like with Bluetooth, make sure you can receive signals inside the case. Also make sure you include the camera board and have the camera pointing outside the case. When you have all the equipment on hand, attach everything together, no case needed, and then run the Pi to a standard monitor and keyboard. Use the larger form factor to configure the Pi. Once everything is configured, shut the Pi down and replace the standard monitor with the touchscreen. Use the touchscreen to navigate and ensure everything is working right. Once it is, it is time to build the tablet.

Assembling the hardware and mounting it all in a case is only the last step. Once it is built up, you have an inexpensive device. Despite its relative low power, it will attach to the internet and allow you to brows freely. With text editors and other document formatting utilities, the Pi will be filled with programs that will make it highly productive. And as you have Python on the Pi, you can even write a few simple games to play on the tablet as well.

2. Know where you go with the Power of the Pi

So Pis are not super, ultra powerful PCs. They are more low-power, but that doesn’t mean they cannot do amazingly powerful things. Among the best things they can do is track locations via GPS. And all it takes is a little bit of know how, some hardware and a Raspberry Pi. By attaching a GPS module and a 3G or LTE modem to the Raspberry Pi, you can track locations. And that’s fine if you want to sit at home and not watch Google maps update with a new location at all. So to get into the field and go mobile with the Pi, we will need some portable battery power.

Unlike many of our other projects, this requires no coding, except to attach your location data to Google Maps. As an interesting side project, you could create an MySQL DB on the Pi and use the device to write data to the DB at regular intervals. Now in addition to the GPS location data being read onto the Pi, now it can be stored to generate a breadcrumb trail when integrated to the maps API. And if you were really clever, you could post updates to a travel blog by attaching the Pi to your vehicle and feeding power directly into it from the battery. Of course, make sure that the Pi powers down when the car turns off, otherwise you will drain your car’s battery and then your trip will be halted until you can get a jump.

Maybe attaching the Pi to the vehicle wasn’t that bright of an idea.

3. Document your travels with an on dash camera

So we’re going to stick with the mobile Pi for at least another project. This one allows the Pi programmer to rely on a hardware only hack to create a simple and potentially lucrative means of documenting what happens as you go on your way. In some foreign countries, thefts and other crimes are so commonplace that citizens have started sticking cameras on their dashboards so as to not fall victim of some scammer who claims that to have been the victim of a hit and run accident. That prevalence of dashboard cameras enabled many to capture amazing video of rarely seen events, like meteors streaking across the sky.

Well, we aren’t expecting some slipping Jimmy to run a scam on us. But we do want to capture cool video. And even better, it would be fun to stream this video if we wanted to. So let’s merge the 3G or LTE modem with the Raspberry Pi and then attach a webcam to it. All that’s left is to sort out how to mount the mobile Pi webcam device and power it up. In this case, we should use some kind of containment box. Once the box is built (or 3-D printed, of course) you can mount it using the hardware you would use to mount an iPhone or GPS device. Then power can be run from one of the 12V power jacks within the vehicle.

An added application for this type of project. If you are using the camera board, or using the Pi to drive an external camera, you can also set the camera up to take pictures as set intervals. Much like using a GoPro to capture a time lapse video of your drive, this is a simple and easy way to get footage of the road while you’re traveling and without the need to fiddle with a camera or other equipment.

4. Listen to the books you don’t have time to read on your Pi

With the rise of sites like audible and the bonanza of available podcasts, more people are listening to the spoken word than ever before. And dozens of options abound for the pleasure of hearing your favorite book or show. But what they all have in common is that none of those options are just for the listening to of audio books or podcasts. As a result, these other devices will frequently hang, stall or crash. I’ve had it happen enough on my phone that I was ready to pull out my hair and never attempt to listen to spoken word media again.

But the Pi saved me. I discovered a project centered around the preservation of the spoken word. And it was to my benefit to sort out the details so that I could get a device dedicated to my favorite way to pass my driving days.

So for hardware, let us assemble one Raspberry Pi, one enclosure, preferably a modular styled one that you can alter to suit the needs of this build, one button to depress to stop or start the audio, speakers, a USB drive (for loading and unloading the audio), an on-board 8GB SD card and some assorted electronics like resistors, LEDs and wire. You can pull the original authors python audio script off of github. The other software elements include the ubiquitous for us Raspbian OS, the music player daemon or mpd, mpd-python, the music player controller or mpc and pyudev. This audiobook player is set to be a dumb device that boots and resumes playing wherever the device left off. The USB drive is used to swap out audio. So if you are going with podcasts, you wills till need a pod catcher and the patience to port them over one at a time when the other finishes playing.

But without a doubt, this simple to use, easy to make and inexpensive to build audiobook player is a surefire way to encourage those who are very busy to take a little time to get a book going and then listen as they go.

5. Wear your Pi

So the Apple Watch debuted to much fanfare and it now dominates the wearable technology marketplace. But unlike those Apple aficionados who love their gear and are willing to stand in long lines for hours and hours to get it, you can have your own, unique, wearable tech built on the Pi.

Wearables are not truly a new category. The Fitbits and other fitness trackers are not so different from pedometers and other wearable fitness technology from the last 20 years. Likewise, the Galaxy Gear, Pebbles and Apple Watches are built on the same basic backbone of your run of the mill digital watch. The only major difference is that these pieces of technology can do more. That’s because they tie to a smartphone or other device. That’s also because they have a highly powerful, mobile computer at their heart. And we have been exploring a highly flexible mobile computer for the last several pages. So the natural conclusion is that we can wear our Pi. We must wear our Pi. And we can even use a USB Bluetooth dongle to attach our Pi to our phones. In fact, some Pi fans discovered a means of getting Google Glass to communicate to hardware and to the Internet as a whole via the Pi. But even beyond that, it’s simple to slap a keyboard and a Pi together and add a small display. They can be linked together with old fashioned means like leather or canvas and be worn on the wrist/forearm area. Just keep in mind, that for you to get any use out of your Pi, have an easily swappable battery back handy.

6. Create your own digital personal assistant

The introduction of Google Now and Apple Siri and now Amazon’s Alexa was meant to herald a new age of voice recognition. And since then Siri has gotten a little bit better at knowing what I want, but not much. The reality is these devices are frustrating because the technology is new. And rather than invest heavily in the many different gizmos and gadgets that will enable you to speak commands into the air and allow your digital personal assistant to respond with something that’s not quite right, why not build your own. And yes, of course, the Pi is the heart of this build, because that’s how versatile Raspberry Pi is.

For this we will need the Pi, a USB power adapter and one USB microphone. The software at the heart of the Pi’s assistance is called Jasper. Once it is installed, you will want to take a little time to attach various other accounts, like your Gmail or Spotify or what have you. Once you have configured your accounts, you are ready to speak your commands to Jasper. Because Jasper is open-source, much like the Pi, it is free to install and use. You can also expand Jasper’s functionality by using Python to write calls to Jasper’s APIs. As new modules are developed, Jasper will grow smarter and more functional. But of course, don’t try to ask Jasper how to bake a Raspberry Pi.

Chapter 7 – Expanding the Capabilities of the Pi

We have covered many practical angles in developing the Pi to its fullest. Now we are ready to cover some ground we had not previously tread. Unlike many the developers of most ordinary average computing platforms, the creators of the Raspberry Pi encourage users to make modifications and truly build out the platform, far beyond their intent or design. As a result, many users do things that don’t match up with regular expectations. We’re going to review a few of them in this chapter and then turn our attention to the fun that we can have with our Pi.

1. Data logging solar energy with the Pi

This is definitely far a field from our regular practical projects. However, this project is built on the reliable constant at the heart of the Pi. Thanks to the presence of the digital input output sensors on the device, the Pi is capable of reading data from most any source. In fact, even analog data signals can be converter through an analog to digital converter. For the particular project that spawned this writeup, the originator used a collection of sensors throughout his home to feed data into the Pi and then publish that data to a website. He also attached a multi-channel analog to digital converter to the Pi to ensure all the signals were logged.

The Pi is logging signals from several solar panels and monitoring the charge from the panels. The Pi is also tracking the voltage feeding into the storage batteries on-site at the home. Temperature sensors track internal temperature as well as the temperature of the home’s water cylinder. This project illustrates that no matter the complexity of the data retrieved and fed into the Pi, the Pi is capable of tracking it, logging it and displaying it.

One of the logical upgrades to this particular project would be to store and log the times of various readings within an attached database. The great big advantage of these kinds of projects is that they can enable an individual to make logical improvements to the systems within one’s home. These improvements are accomplished through collection and analysis of data. That’s what the Pi does rather well.

2. Automating the preparation of food

If you have never heard of the process of food prep called sous vide, you have successfully avoided the foodie movement. But if you know what sous vide cooking is and have experienced the blissful results of rough cuts of meat turned into succulent delicacy thanks to the amazing poaching of the sous vide gear. Sous vide cooking relies on two essentials. The first is a consistent water temperature.

Because water is denser than air, foods can cook quicker in water that has been elevated to a specific temperature. The transference of heat energy from the water into the food is accomplished by the utilization of sealed, air-tight bags. Those bags adhere directly to the surface of the food. Because the bags will not allow water to come in contact with the food, the food's flavor and texture are preserved. Because they are immersed in the water, they cook faster and more consistently. It's a near ideal means of preparing food.

But (and there is always at least one but, isn't there?) the cost to obtain sous vide rigs is prohibitive. When the average at home culinary engineer spends less than one hundred dollars on appliances, the sous vide's which clock in between several hundred and several thousand dollars, are luxury items.

The average cook's range and oven cost less. Some folks have the budget to blow on high-end cooking gear. But the average person is in need of an alternative solution. Enter the Raspberry Pi. This Pi project requires a very low-tech device – an electrical crock pot. The slow cooker will maintain the consistent temperature necessary to ensure even cooking and moist, tender meats. This is due to a fairly simple feedback loop that begins in the pot and cycles data back to the Pi which will adjust the power hitting the cooker. The end result is a sous vide system at a fraction of the cost.

3. Pi technology drives greater creativity across media

We are confronted nearly every day with ads. The ads we see show up on our search engine results and in our feeds of social media feeds. But one of the aspects of creativity with advertising that is driven through the improvement of technology is the introduction of new and innovative means of conveying information to people. But the key way that advertisers try to do this is through the use of technology to catch people's eyes. This is nothing new. Color televisions supplanted black and white sets not because they produced a better picture, but because advertisers found that color sets helped ads get better play. Likewise, high-definition television sets are better advertisers than standard definition sets. And photo illustrated advertisements for specific products have a stronger impact on the public than those illustrated with art, no matter how talented the artist's rendition of the product might be. So when a Scandinavian ad agency attached a collection of Raspberry Pis equipped with ultrasonic sensors to video billboards in a train station, the results were eye-catching.

The ultrasonic sensors determined when the train was arriving in the station. The receipt of this data in the Pis triggered adjustments to the video. Instead of the model in the video standing with still, straight hair, her hair moved to reflect the type of wind gusts that would accompany the arrival of the train. The video boards were updated by the Pis as they heard what was going on around them. The sudden shift of the video is both eye catching and innovative. In a culture that has been so saturated with ads that most of us tune them out by instinct, this kind of application will draw and hold attention, which is exactly what advertising is meant to do.

4. Kick the Pi into higher gears

Among the biggest joys of the Pi is that its inventors encourage innovation. They also encourage dialing things up some. Because they encourage overclocking, this has given Pi users plenty of options to ensure they get the performance they need. The old AMD Athlon chips were also great for overclocking, but doing so would void the warranty. And that was the case with the Raspberry Pi at first. Overvolting the chips was possible, but it increased the likelihood of system failure or decreasing the mean time before failure for the system on chip at the heart of the Raspberry Pi. But with one release, everything changed. The foundation that creates the Raspberry Pi launched a turbo mode that made it possible for users to get increased performance, without blowing out the chip.

Helpfully, the Pi includes preset overclock settings that can take the processor speed from the baseline of 700MHz all the way up to 1GHz. And that uptick in performance is reflected throughout the standard benchmarks of computing performance. As part of the release, the foundation also introduced the ability to monitor CPU core temperature to ensure it did not get close to the maximum temp before the core began to melt. This effort put more power into the hands of users. While this isn’t a project in the same sense as out other projects, it is very much a factor that makes the Pi revolutionary in the marketplace. As far as I know, there has never been a platform developer more supportive of taking the device to extremes.

5. Make a daisy chain and make the Pis more powerful in the aggregate

Professors at the University of Southampton in the United Kingdom found a means of further kicking the computing power of the Raspberry Pi. They daisy chained 64 total Raspberry Pis together to discover if connecting multiple systems would enhance processing power. The biggest challenge the team faced was getting their hands on enough Pis to proceed. Considering the foundation thought they would sell a few handfuls of Raspberry Pi computers, and they have sold more than two million is a testament to how challenging fulfillment was at various points in time. They used Debian, but the build of Debian was the same as the standard Raspbian that we have been using on most all of these projects.

Amusingly, the team at Southampton used legos to create racks into which the Pis were inserted. These racks kept the costs minimal as they strung the system together. The individual Pis were connected via a message passing interface. Their initial test was to calculate Pi on the Raspberry Pis. The team concluded that the low cost of the Raspberry Pi made it ideal for this type of processing application.

Chapter 8 – The Fun Side of the Pi

We’ve gone deep into practical uses and we’ve lingered over the places where the Pi can be overclocked and stretched beyond its design parameters. Now we get down to the fun things that Pi owners can do with their Pis. This is where creativity becomes a factor in the builds and the projects diverge from function and form into fun. So with that out of the way, let’s begin with the age old practice of men and boys of all ages – remote controlled vehicles.

1. Tank columns directed by the Pi

Remote controlled vehicles are beloved among hobbyists. They may go by the name of drone today, but really they are just RC units. But what makes the tank project different is that the tank is not run by regular remote. It's being driven from instructions that are being fed to it via a Raspberry Pi. And those commands are originating in the Secure Shell. The hardware needed is the Pi itself and a USB WiFi dongle. The WiFi attaches the Pi to the wireless network upon boot up. From there, the Pi automatically shoots commands down the Secure Shell into the device.

For this project, the real trick of it is being able to direct commands directly from the Raspberry Pi into the tank itself. Taking the tank itself apart to allow for manipulation of the remote is a significant challenge. But that difficulty is overcome by the sense of accomplishment as the tank rolls down the sidewalk. This intrepid inventor (no other title for him) attached scopes to the tank to read how signals were received. From their he was able to determine what each signal was doing and how to replicate those signals using a transistor to shield the general purpose input output chip on the Raspberry Pi.

At that point, all that was left was to enable the Raspberry Pi to draw power directly from the tank's batteries. The actual steps exceed 30 individual processes. The level of effort makes cracking the tank prohibitive, but it does prompt me to wonder what is possible with remote controlled vehicles. If one determined developer can get a tank driven from a Pi, what else can we use the Pi to control?

2. How about helicopters?

Yeah, so now we truly are getting into the drone side of the pool. That's because spinning up a quadcopter is akin to the application of drone technology currently in place for military and commercial uses. These types of remote controlled vehicles are increasing in popularity. The initial uptick in interest could be attributed to military use. But increasingly the private sector is getting in on the quadcopter game. Now with the Pi based vehicle, hobbyists can deploy their own army of remote controlled quadcopters.

Among the obvious applications is the utilization of on-board cameras to scope out territory. I think it is only a matter of time before these drones hover over the various fields of play that dominate weekends on the television dial. Marry one of these to a GoPro that is feeding 4K video at 30 fps to a control room and you have unique perspectives on the sporting pastimes that tend to dominate the American culture. And as we noted in the last chapter with regards to the advertising industry, creativity is driven by technological advances. The same is true for media. That's in part because media is driven by advertising.

With quadcopters capable of being produced en masse and distributed with the Raspberry Pi controlling them, the future applications for amateur drone activity are literally unlimited. You can find the code used to control the copter on github. Just remember, the Raspberry Pi is considered low-powered compared to the processors associated with computers. But when compared to the computing power on remotely controlled vehicles, the Raspberry Pi kicks tail rotors. As such, there is plenty of power to feed commands while the Pi is flying the copter. The next step would be to use the Pi to manage other devices, such as cameras or the transfer of streaming video to other systems while receiving and executing commands.

3. You may be full of hot are, but the Pi is not

Hot air balloons have been used to transport people overland for several centuries. Even so, the primary application for ballooning was reconnaissance. And with the introduction of high altitude ballooning, the ability to send light cargo to a distance of 18 miles above the surface of the earth has opened up whole new frontiers to amateur and professional astronomers alike. What truly makes high altitude ballooning remarkable is the unprecedented access to pictures from these unbelievable heights.

When the balloons go up, they are typically carrying a payload that includes sensors and GPS and perhaps a camera that is writing images to a SD card. When the balloon achieves a certain height, the payload drops and trackers use a feed of data on a low power radio link to determine the location of the drop. The Raspberry Pi was first thought to be ill-suited as a computer for high altitude ballooning flights. Mainly this perception was fed by the weight and power draw of the Raspberry Pi. Again our perceptions will startle us if we let them. We think the Raspberry Pi is a low-power, lightweight system. And in comparison to our standard PCs, they are. But compared to other systems on chips, the Raspberry Pi can be down right bulky and a power hog. When the max payload is a little more than two pounds, the difference of a few ounces is literally make or break.

But what the Pi has that other boards lack is the ability to attach a webcam to deliver pictures down the radio link that normally carries telemetry information. Packing the Pi in cushioned material can help it and its on-board data storage survive the drop from 18 miles up. That makes the Pi viable for near space exploration.

4. Let your Pi make you money by mining for… Bitcoins

In recent years, American culture was overwhelmed by the rapid ascent and equally quick descent of Bitcoin value. The cryptocurrency captured many people’s imagination and filled their heads with thoughts of wealth untold. The problem was that few understood how to obtain Bitcoins. Fewer still understood how capriciously the currency was valued. As such Bitcoin boomed and busted and broke the hearts of speculators who got in right before the bottom dropped out.

But that doesn’t mean the value of Bitcoins vanished entirely. As the crypto currency releases new tender on a consistent basis, the utilization of computers to “mine” for Bitcoin is a fairly common practice. Though the Pi is lower powered than the currently available systems on the market, it is a low-cost and affordable means of seeking out Bitcoins as they become available. Since the crypto currency’s value is driven largely by demand, having a platform to mine for newly minted Bitcoins will allow a user to steadily build up value. While the value will be low initially, the value increases as demand for Bitcoins grow. And should the market approach the peaks of recent memory, even a small supply of Bitcoins can turn a handsome profit.

Chapter 9 – More Fun Pi

We have a few more fun projects to spark your imagination. These fall into the whimsical or fun side of things. Nothing like tanks and copters and balloons headed for near space orbit, but certainly things that highlight creativity and the enjoyment of life.

1. Let Pi serve up your pics

So you have a lot of pictures. We all do. The ubiquity of digital cameras and the every declining costs of solid state memory make photography a cheap hobby. But if youa r elike me, you have a bunch of pictures alls tored on a hard drive that you never look at. You took the time to take them, but you never take the time to look at them. Wouldn't it be great if we had a means of displaying pictures. Like a picture frame, only with all the pics you've taken.

Of course, they do exist -- digital picture frames. And they are nice. And fairly inexpensive. But with the right mix of skill and know how, you can build your own custom picture frame, with the exact display size you want and plenty of memory all by using your Raspberry Pi.

To accomplish this, we need first to find the right screen. To be brief, we want to have a large screen. Commercialy available digital picture frames typically run around 8 inches diagonal. Bit with LCDs relatively inexpensive, we can get a high-end 15" or 17" LCD. Go ahead and order a laptop screen so you don't have to worry about a bezel. But keep in mind, buying a regular laptop screen is an invitation to the next thing we need to buy.

You will want to take extra care in selecting an LCD control board. These printed circuit boards allow for video signal to be fed to the display. The best boards cost relatively little and come with an inverter, control switch and the right connector cable. Quality merchants will also make sure the board is programmed to work out of the box.

Once you have the LCD and controller, you can take measurements for a frame.

A lot of people take the measurements off the LCD vendor's site as accurate. They end up having an extra frame lying around, too. Wait until it arrives, measure twice and buy once. When the frame arrives, we can also look at mounting options.

Then it is time to buy the Pi. Order the Pi plus an SD card for the OS, a USB WiFI adapter, a USB power adapter, and a case if you need or want one.

When the Pi arrives, prep it the same way we did with the multimedia player project. Raspbmc is going to be our media player of choice again. Once the Pi is prepped, we build. Start with the LCD controller board. Then wire everything up and move to the frame. Once your frame is built, we will need to setup the media on the Raspberry Pi. And once it is all loaded on, the next step is to enjoy the beautiful photos you've taken on the beautiful digital picture frame you've built.

2. Yule time treats are made better by Pi

Even before Clark W. Griswold dedicated the Griswold Family Christmas, husbands have been scaling their homes to put up Christmas lights. And like Clark, they have been putting forth one part of the effort. The visual display catches the eyes of passersby. But what will make them stop cold in their tracks is the multimedia effect of lights and sounds. Play the songs and queue the lights to correspond and you have a professional grade Holiday extravaganza.

The Pi is uniquely suited for the process of lights. Thanks to the on-board input output controllers, the lights can be keyed to go on or off directly from the board. In the not to distant past, hobbyists used plug-in digital I/O boards to control their lights with a software program, which frequently they had to write themselves. The Pi will even allow them to continue running the same program they had before.

So the lights are driven off the input output controller. How then do we use the music to key the lights. Thankfully, the Pi includes a Midi sequencer that will route commands embedded in the music to a particular port. As long as the I/O controller is listening to the port, the lights will fire in sequence.

With that we are ready to build. We will need a Raspberry Pi and case to start. We also need a five-volt solid state relay board. They can be had realtively cheaply. You will also want a handful of female to male gender changers, some electrical outlets, boxes and cover plates. You will also need some non-metallic wire and some electrical cable, figure fifty feet of each. If you are planning to mount lights on plywood rather than on your home, get plenty of plywood, too. A small amplifier and a speaker system with speakers will make sure the music can be heard.

For software we being with Raspbian. We also want Timidity and the Alsa development libraries for the Midi files. Finally, we need Gordon's wiringPi library. You can find and download the source code for the light control by searching for Christmas lights and Raspberry Pi through Google. Once the software is compiled the ports configured and all your lights wired up, you are ready to fire up the Pi and make the Christmas season remarkable.

3. Put your stuff in the cloud via the Pi

Dropbox and other services allow users to store files in the cloud. But way back when we talked about VPNs and Tor and other privacy measures, we realized that putting your data into someone else's control is a bad, bad idea. So we can overcome that flawed decision, by turning our Pi into a personal cloud server.

The hardware for this project is very similar to the project we did for the network attached storage. We will need a Raspberry Pi, one USB hard drive and a USB WiFi adapter.

Our first step is to assign a static IP address to the Pi. Following the assignment of the IP, let's update and upgrade the Pi. And when that completes, we need to install Apache with some options. We want SSL, PHP APC and PHP5. Once those are installed, all our software will need to be configured. And when you have finished configuring the software you must restart Apache. Now we can install own cloud.

Once own cloud is installed, it will also need to be configured and the drive will need to be partitioned and formatted. Upon completion of all the configuration, own cloud will be ready to accept your files, which will live in the cloud and be accessible from any device that can log into the server.

4. Beginning robots with Pi

So we talked about the drones, which remain essentially human controlled. But this little project, which was featured on the Raspberry Pi foundation website is one that gets us closer to real Pi robotics. This inventor stated with an old Roomba. He stripped the brushes and motors attached to the brushes out. What was left were the drive wheels and the coaster wheel along with the unit's base.

The drive wheels retain the necessary transistors to deliver power to the motor so we need not worry about using the Raspberry Pi to power the drive wheels. We do not need an expensive relay board. We need only connect the bases of each transistor to the GPIO. The inventor elected to directly power the Raspberry Pi using a power supply circuit. Attaching a webcam will allow the robot to shoot video as it moves.

The inventor left his custom code up on the Raspberry Pi site. In addition, he explains how he setup a Wii remote control to guide the robot. It’s an interesting project and shows how easily robots can be created to run with the Raspberry Pi.

Chapter 10 – The Business Side of the Pi

I alluded to this idea early on, when we first started exploring practical uses for the Pi. And plenty of entrepreneurs have taken the plunge into building products for consumer use based on the incredible, inedible Raspberry Pi. So while we've begun to see business ideas centered on the Pi arrive, it's not too late to develop your own. The benefit of working with the Pi is the price. You cannot beat the low cost of entry for a fully functional computer platform.

Thankfully, the creators and promoters of the Raspberry Pi recognize that this is agoodness. They do not require extensive and expensive licensing for their product. All they request from product developers and inventors is a little attribution. So we have affordability of hardware cost and licensing. With that we can build up products that satisfy the most important criteria for success. First, the product must solve a need or facilitate the attainment of a want. Second, the product must be inexpensive to make. Third, producing the product must be scalable, with an easy to develop back end. Finally, the product should also lend itself to complementary products that can be rolled out to the existing customer base.

Because the folks at the Raspberry Pi foundation see the benefit of developers using their platform, you as an entrepreneur already have a business partner. That partner is committed to your success and will share in the rewards of your endeavor. But rather than taking the big bite out to allow you to use their platform in a licensing deal, like many foundations, they allow their product to be used freely. When you buy it you can develop as many products as you like. And as an added bonus, when you come up with something brilliant, the good folks at the foundation will help you out by promoting your product on their digital pages.

As an entrepreneur, you could not ask for a better technological partner. The benefit is that they will continue to refine and improve the platform your product is built on. Therefore, you can focus on making the code ideal. So brush up on your python programming skills.

However, the Foundation does not explicitly endorse products. There is a vast number of developers who are building useful products to distribute based on this platform. And with the knowledge you have already gleaned from this book, you should be able to join them. So the balance of this chapter is devoted solely to the exploration of what you need to do.

Product development begins with an idea. You have forty or so of them by now, and there is a very good chance you have thought of some that I haven't mentioned.

From that idea, a successful inventor looks at how he or she can best turn that idea into a solution that enables people to solve a problem with it. So let's talk print servers. With the details from the third chapter, you can create a simple interface to print from any network connected device, whether WiFi or hard connected to the network. For people with several devices a simple, inexpensive print server would address a critical need. They can work on things on any device, print from anywhere and never need to worry about plugging in a printer. Now, HP and Dell offer those kinds of printers, but they cost more for a network interface. Now you have a means of turning an inexpensive printer into a network printer. For a user who doesn't want to bother buying a new printer, this is ideal.

With an idea that solves a problem, the developer then executes. The first step is building the prototype. Then users will test it and provide feedback. Finally, a successful proof of concept is ready. The product developer then designs the housing. At which time, the product is ready to go to market.

A successful developer knows this is where it gets tricky. From this point, the developer needs to raise awareness, drive demand and secure distribution. Awareness begins with product marketing. Modern technology has helped with this. A successful entrepreneur will turn to crowdfunding sites and social networks for initial promotion. Start a funding campaign and offer a first run product to contributors. Promote this campaign to your friends and acquaintances on Facebook, Twitter, Instagram and Pinterest. They will often pass along this campaign to their friends and acquaintances that you do not know.

Through the social proof of your friends' endorsement, you are garnering your first customers. The folks are signing onto the validity of your idea. And as such, you can charge a slight premium on the eventual market price to get startup capital. Getting capital is only one benefit. You can also correctly estimate the size of your initial production run.

An entrepreneur will also use different funding tiers to distribute promotional products. A first tier contribution – say, ten bucks or so – gets a sticker with the company or product logo that can be stuck on a laptop. A second tier contribution will get the promotional sticker and maybe a coffee mug with the logo or some other more tangible gift. The third funding tier is driven off the desired retail price of the product. Contributors at this level will receive the product itself as well as the second and first tier products. The nice part is that even if someone gives at a lower funding tier, they have become a customer. You can market the product to them through email newsletter or direct postal mail appeals. More importantly, you have made sales. So get to work on fulfilling those orders.

With awareness raised, we need to next set up a fulfillment system. Product distribution begins with you taping boxes and bringing them to the post office. But that is not effective long term. A great alternative is to set up a Fulfilled By Amazon account and send your exclusive product to a company that handles logistics like no other. Amazon will handle fulfillment and take a bite out of your profits. But this is merely a cost of doing business. The goal is not to spend forever managing your business. It is to get the most efficient means of connecting your product to consumers. The funds you raise from your campaign will cover the initial costs of having Amazon warehouse your product and fulfill your orders.

Finally, we need to create marketing campaigns to drive demand. The options here are plenty and varied. Send a product to tech journalists to review to garner press coverage. Use social media pages and ads run on Google search or Facebook to drive people to your business. With fulfillment squared away, every order represents a return on your investment of time and effort to create a compelling product.

Keep in mind, product development is not the only way to make money from the Pi. In fact, the simplest way involves no coding expertise at all. You can distribute Raspberry Pis to consumers who would not otherwise be able to buy them. An inexpensive computer will always be in demand. The foundation's current distribution partners are Anglo-centered. As such, many potential customers outside the United States have no access to the Pi.

Where demand exists, a savvy businessperson steps in to meet the need. In doing so, he or she makes a tidy profit that can fund further projects. An entrepreneur can snap up Raspberry Pi computers at $35 each and ship them to retailers who mark them up to a higher price tag. Those retailers can provide their customers a full line of accessories that generate more profit.

This is a form of retail arbitrage. When suppliers are not serving customers, their demand drives prices higher. The higher price encourages more suppliers to enter the marketplace. This process ensures the best allocation of product.

Raspberry Pi is a low-cost, technological breakthrough in embedded computing. That ensures demand will be high across the globe. Building distribution out to other countries will bring substantial profit. Entrepreneurs who want start up capital to build a Pi based product can earn it through distribution. Like above, that process will create new customers for the Pi based product in the future.

As you can see, this platform is ideal from business ventures. Improving your coding skills is only part of what you need to do to ensure success. You will also need to improve your business acumen and look for ways to maximize your profit from every available avenue. In doing so, you will find more than just success. You will also develop the skills necessary to run a profitable and successful business.

Conclusion

Your interest in the amazing Raspberry Pi shows me that you are looking for a new challenge. In studying it and Python, you are opening the door to several ways to create useful products for either your own use or to sell to an eager public who will benefit from your vision, creativity and ingenuity. So let’s recap.

Python is a simple language to learn and an easy one to work with to create simple hacks or complex gadgets and gizmos. The beauty of combining an effective programming language with an inexpensive, powerful embedded computing platform means you can create virtually anything you want. And it just takes basic mastery of a simple programming language and some easy to obtain hardware.

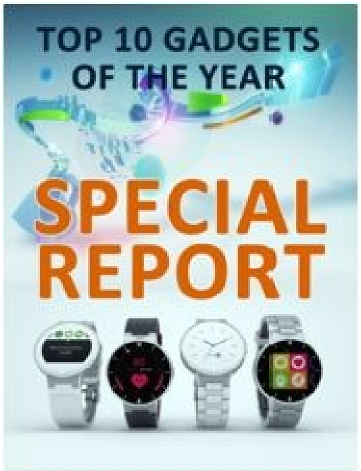
For fledging inventors, Raspberry Pi creates the ultimate entry level solution to build a basic gadget. Its low cost makes the gadget affordable. Then once you have created the code you can begin to market and sell these devices. They may not change the world, but they create the credibility to show potential business partners that you can take an idea from inception to completion. And that ensure whatever you want to do will be funded and supported and successful.

When you have mastered the art of designing products using the Raspberry Pi platform, you can next turn to mass production and distribution. Remember products need to solve a need, provide value for their price and be simple to produce and ship to customers. Build your initial sales through crowdfunding to build up the capital resources you need to mass produce your product.

Alternatively, look to distribution of the Raspberry Pi itself as a means of getting money into your business to fund development of new products.

As you have seen, the practical applications of the Raspberry Pi are only limited by your own creativity and imagination. So the more you think on possible problems to solve and the more you think outside the box to solve those problems, the more successful you will be. I hope you have enjoyed learning about Python and Raspberry Pi and that all your projects and products are tremendous successes.

FREE Bonus Reminder



If you have not grabbed it yet, please go ahead and download your special bonus report “*TOP 10 Gadgets Of The Year*”.

Simply Click the Button Below

OR Go to This Page

http://bit.ly/1AiwLLM

BONUS #2: More Free Books

Do you want to receive more Free Books?

We have a mailing list where we send out our new Books when they go free on Kindle. Click on the link below to sign up for Free Book Promotions.

=> Sign Up for Free Book Promotions <=

OR

Go to this URL http://bit.ly/1COlFPe