MicroPython for the Internet of Things

A Beginner's Guide to Programming with Python on Microcontrollers

—

Breaking the C-like language barrier to make device programming easy and fast

—

Charles Bell

Apress®

MicroPython for the Internet of Things

A Beginner's Guide to Programming with Python on Microcontrollers

Charles Bell

Apress®

MicroPython for the Internet of Things

Charles Bell
Warsaw, Virginia, USA

ISBN-13 (pbk): 978-1-4842-3122-7 ISBN-13 (electronic): 978-1-4842-3123-4
https://doi.org/10.1007/978-1-4842-3123-4

Library of Congress Control Number: 2017960889

Cover image designed by Freepik

Managing Director: Welmoed Spahr
Editorial Director: Todd Green
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Technical Reviewer: Nelson Goncalves
Coordinating Editor: Jill Balzano
Copy Editor: Karen Jameson
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit http://www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484231227. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

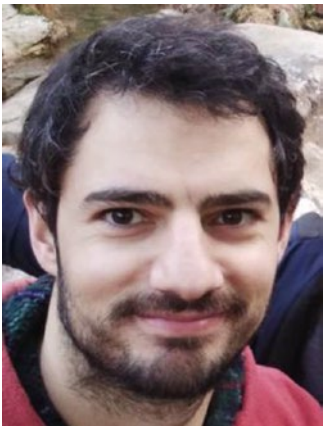
I dedicate this book to my big sister for instilling in me the thirst for knowledge.

Contents

About the Author

Dr. Charles Bell conducts research in emerging technologies. He is a member of the Oracle MySQL Development Team as a Senior Developer assisting in the development of MySQL high availability solutions. He lives in a small town in rural Virginia with his loving wife. He received his doctorate of philosophy in engineering from Virginia Commonwealth University in 2005. His research interests include database systems, software engineering, sensor networks, and 3D printing. He spends his limited free time as a practicing Maker focusing on microcontroller and 3D printers and printing projects.

About the Technical Reviewer



Nelson Goncalves is a member of the MySQL Developer Team at Oracle, currently working with the InnoDB cluster Team. He is based in Portugal and working for MySQL since 2013. Before joining MySQL, he was an MSc student at Universidade do Minho where he specialized into Formal Methods and Distributed Systems. He loves all things Python related.

Acknowledgments

I would like to thank all of the many talented and energetic professionals at Apress. I appreciate the understanding and patience of my editor, Jonathan Gennick; and managing editor, Jill Balzano. They were instrumental in the success of this project. I would also like to thank the army of publishing professionals at Apress for making me look so good in print with a special thank you to the reviewers for their wise counsel and gentle nudges in the right direction. Thank you all very much!

Most importantly, I want to thank my wife, Annette, for her unending patience and understanding while I spent so much time with my laptop.

Introduction

Internet of Things (IOT) solutions are not nearly as complicated as the name may seem to indicate. Indeed, the IOT is largely another name for what we have already been doing. You may have heard of "connected devices" or "Internet-ready" or even "cloud-enabled." All of these refer to the same thing — be it a single device such as a toaster or a plant monitor or a complex, multidevice product like home automation solutions. They all share one thing in common: they can be accessed via the Internet to either display data or interact with the devices directly. The trick is applying knowledge of technologies to leverage them to the best advantages for your IOT solution. In this book, we explore how to build IOT solutions using an easy-to-understand programming language named MicroPython running on small, dedicated microcontroller boards.

Intended Audience

I wrote this book to share my passion for Python and IOT solutions. I especially wanted to show how anyone can program their own IOT solutions in Python using MicroPython on small microcontroller boards. The intended audience therefore includes anyone interested in learning how to build IOT solutions, hobbyists, and enthusiasts who don't want to spend a lot of time learning a complicated programming language to control hardware through software in IOT solutions.

How This Book Is Structured

The book was written to guide the reader from a general knowledge of microcontrollers and MicroPython to expertise in developing MicroPython solutions for the IOT. The first several chapters cover general topics including a short introduction to the Internet of Things, what microcontroller boards are available as well as how MicroPython works. Later chapters present a tutorial on programming in MicroPython as well as an introduction to electronics. This is followed by four projects that you can implement to learn how to build MicroPython IOT solutions. Throughout the book are examples of how to implement many of the concepts presented. The following is a brief overview of each chapter included in this book.

- *Chapter 1, "What Is the Internet of Things?"*: This chapter presents and answers the questions of what the IOT is and how IOT solutions are constructed. You are introduced to some terminology to describe the architecture of IOT solutions as well as some examples of well-known IOT solutions. The chapter concludes with a demonstration of MicroPython.

- *Chapter 2, "Introducing MicroPython"*: This chapter presents an overview of what MicroPython is and how you can get started using MicroPython boards.
- *Chapter 3, "MicroPython Hardware"*: This chapter discusses some of the hardware available for MicroPython including the micropython.org (Pyboard) and Pycomm (WiPy) line of microcontroller boards and several other alternative boards. The chapter also presents some of the accessories available for each board.
- *Chapter 4, "How to Program in MicroPython"*: This chapter presents a tutorial on learning to program in MicroPython. It covers all of the basics of the language you need to get started writing your own MicroPython scripts.
- *Chapter 5, "MicroPython Libraries"*: This chapter presents an overview of the various MicroPython libraries available for use in your scripts. It includes many examples of how to get started using the libraries to interface with hardware.
- *Chapter 6, "Low-Level Hardware Support"*: This chapter presents an overview of the low-level hardware abstractions available for the Pyboard and WiPy ports of MicroPython. The differences of the libraries are presented along with several complete examples to demonstrate the functionality.
- *Chapter 7, "Electronics for Beginners"*: This chapter presents a short introduction to electronics including the types of components you will be using in the book along with a list of recommended tools. The chapter concludes with a survey of the types of sensors available for IOT solutions.
- *Chapter 8, "Project 1: Hello, World! MicroPython Style"*: This chapter presents a hands-on project to help get you started programming hardware and building MicroPython solutions. The project is a clock programmed in MicroPython using a real-time-clock (RTC) module.
- *Chapter 9, "Project 2: Stoplight Simulator"*: This chapter presents another hands-on project that interfaces with LEDs and buttons to build a pedestrian stoplight simulation. The project also demonstrates how to control your hardware remotely via a web page.
- *Chapter 10, "Project 3: Plant Monitoring":* This chapter presents a more complex hands-on project that demonstrates how to generate sensor data and view it over the Internet. The project is a plant monitoring solution that you can expand from one to many plants.

- *Chapter 11, "Project 4: Using Weather Sensors"*: This chapter presents the last hands-on project that combines all that you have learned in the book to build a working IOT solution. The project is a small weather sensor node that uses the new Adafruit IO cloud services to store and visualize the data.
- *Chapter 12, "Where to Go from Here"*: This chapter concludes the tour of MicroPython IOT solutions with suggestions for more projects to explore and where to go to find new project ideas including where to look to find answers to questions or problems you may encounter when developing your own MicroPython IOT projects. The chapter also discusses how you can join the community of IOT, MicroPython, and electronics enthusiasts by becoming a Maker.

How to Use This Book

This book is designed to guide you through learning more about what the Internet of Things is, discovering the power of MicroPython, and seeing how to build your own IOT solutions.

If you already have your own MicroPython board and are familiar with some of the topics early in the book, I recommend you skim them so that you are familiar with the context presented so that the later chapters, especially the examples, are easy to understand and implement on your own. You may also want to read some of the chapters out of order so that you can get your project moving, but I recommend going back to the chapters you skip to ensure you get all of the data presented.

If you are just getting started with MicroPython and microcontrollers, I recommend reading the book in its entirety before developing your own IOT solutions. That said, many of the examples presented in the early chapters are building blocks for what follows in the project chapters.

Downloading the Code

The code for the examples shown in this book is available on the Apress web site, `www.apress.com`. You can find a link on the book's information page on the Source Code/Downloads tab. This tab is located in the Related Titles section of the page.

Contacting the Author

Should you have any questions or comments — or even spot a mistake you think I should know about — you can contact the author at drcharlesbell@gmail.com.

CHAPTER 1

What Is the Internet of Things?

If you've been watching the technology world lately, chances are you have encountered numerous mentions of the term, the Internet of Things. Most media references and company advertisements label this or that as the Internet of Things but with little or no explanation about what it means. Even when you do find some depth of what it means, the text tends to focus on either the problems and challenges, or they focus on the promise of making our lives better in the future. Some suggest the Internet of Things will bring about the inevitable evolution of our society as we become more connected to the world around us every day.

However, you need not dive into such heady concepts or recite rhetoric to get started with the Internet of Things. In fact, through the efforts of many open source developers and vendors, you can explore the Internet of Things without intensive training or expensive hardware and software. Best of all, you can explore the Internet of Things without learning a lot about programming or spending months learning how to code!

This book is intended to be a guide to help you understand the Internet of Things and to begin building solutions that you can use to learn more about the Internet of Things. Since this is a beginner's book, we will start by examining the programming language and environment followed by a detailed look at the hardware. We will also learn the basic knowledge of electronics and then explore several projects to help us understand how to work with the software. The final project will bring all the aspects together to help understand what the Internet of Things is, and even how to write custom software for building solutions for the Internet of Things. Best of all, we do so using one of the easiest to use programming languages and easy to use open source microcontroller boards.

So, what is this Internet of Things, hence IOT?[1] Let's begin by explaining what it isn't. The IOT is not a new device or proprietary software or some new piece of hardware, nor is it a new marketing scheme to sell you more of what you already have by renaming it and pronouncing it "new and improved."[2] While it is true the IOT employs technology and techniques that already exist, the way they are employed, coupled with the ability to access the solution from anywhere in the world, makes the IOT an exciting concept to explore. Now let's discuss what the IOT is.

[1] https://en.wikipedia.org/wiki/Internet_of_Things

[2] For example, everything seems to be cloud-this, cloud-that when nothing was changed.

C. Bell, *MicroPython for the Internet of Things*, https://doi.org/10.1007/978-1-4842-3123-4_1

The essence of the IOT is simply interconnected devices that generate and exchange data from observations, facts, and other data making it available to anyone. While there seems to be some marketing efforts attempting to make anything connected to the Internet an IOT solution or device – not unlike the shameless labeling of everything 'cloud,' IOT solutions are designed to make our knowledge of the world around us more timely and relevant by making it possible to get data about anything from anywhere at any time.

As you can imagine, if we were to connect every device around us to the Internet and make sensory data available for those devices, it is clear there is potential for the number of IOT devices to exceed the human population of the planet and for the data generated to rapidly exceed the capabilities of all but the most sophisticated database systems. These concepts are commonly known as addressability and big data and are two of the most active and debated topics in IOT.

However, the IOT is all about understanding the world around us. That is, we can leverage the data to make our world and our understanding of it better.

The Internet of Things and You

How do we observe the world around us? The human body is a marvel of ingenious sensory apparatus that allows us to see, hear, taste, and even feel through touch anything we encounter. Even our brains can store visual and auditory events recalling them at will. IOT solutions mimic many of these sensory capabilities and therefore can become an extension of our own abilities. While that may sound a bit grandiose (and it is), IOT solutions can record observations in the form of data from one or more sensors and make them available for viewing by anyone anywhere via the Internet.

Sensors are devices that produce either analog or digital values. We can then use the data collected to draw conclusions about the subject matter. This could be as simple as a sensor to detect when a door, window, or mailbox is opened. In the case of a switch on a mailbox, the knowledge we gain from a simple switch opening or closing (depending on how it is implemented and interpreted) may be used to predict when incoming mail has arrived or when outgoing mail has been picked up. I use the term predict because the sensor (switch) only tells us the door was opened or closed, not that anything was placed in or removed from the mailbox itself – that would require additional sensors.

A more sophisticated example is using a series of sensors to record atmospheric data such as temperature, humidity, barometric pressure, wind speed, ambient light, rainfall, etc., to monitor the weather that allows us to perform analysis on the data to predict trends in weather. That is, we can predict within a reasonable certainty that precipitation is in the area.

Now, add the ability to see this data not only in real time (as it occurs), but also remotely from anywhere in the world, and the solution becomes more than a simple weather station. It becomes a way to observe the weather about one place from anywhere in the world. This example may seem to be a bit commonplace since you can tune into any number of television, web, and radio broadcasts to hear the weather from anywhere in the world. But consider the implications of building such a solution in your home. Now you can see data about the weather in your own home!

In the same way, but perhaps on a smaller scale, we can build solutions to monitor plants to help us understand how often they need water and other nutrients. Or perhaps we can monitor our pets while we are away at work. Further, we can record data about wildlife in our area to better understand our effect on nature.

IOT Is More Than Just Connected to the Internet

If a device is connected to the Internet, does that make it an IOT solution? That depends on whom you ask. Some believe the answer is yes. However, others (such as myself) contend that the answer is not unless there is some benefit from doing so. For example, if you connected your toaster to the Internet, what could be the benefit of doing so? It would be pointless (or at least extremely eccentric) to get a text on your phone from your toaster stating that your toast is ready. So, in this case, the answer is no.

However, if you have persons such as responsible teenagers or perhaps older adults whom you would like to monitor, it may be helpful to be able to check to see how often they use their toaster and when. That is, you can use the data to help you make decisions about their care and safety.

To me, if there is no use for the data, whether it is something that is viewed in real time or is stored for later processing, then simply connecting it to the Internet does not make it an IOT solution. There must be some gain in the use of the device. Thus, being connected to the Internet doesn't make something IOT. Rather, IOT solutions must be those things that provide some meaning – however small that has benefit to someone or some other device or service.

More importantly, whatever we build IOT solutions to do, they allow us to sense the world around us and learn from those observations. The real tricky part is in how the data is collected, stored, and presented. We will see these in practice through examples in later chapters. See the sidebar for an example of a controversial IOT device - a common household appliance.

However, IOT solutions can often take advantage of companies that provide services that can help enhance or provide features you can use in your IOT solutions. These features are commonly called IOT Services and range from storage and presentation to the infrastructure services such as hosting.

INTERNET-ENABLED APPLIANCES: IOT OR MARKETING HYPE?

One of the ideas or concepts that seems to becoming popular is the connecting of major household appliances to the Internet. While manufacturers may want you to believe this is a new and exciting IOT device, the truth is it is neither a new idea nor is it a world changing IOT solution.

I was fortunate to participate in a design workshop held on the Microsoft campus in the late 1990s. During our tour of the campus, we were introduced to the world's first Internet-enabled refrigerator (also called a smart refrigerator or simply Internet refrigerator).[3] There were sensors in the shelves to detect the weight of food. It was suggested that, with a little ingenuity, that one could use the sensors to notify your grocer when your milk supply ran low, which would enable people to have their grocery shopping not only online but also automatic.

[3] https://en.wikipedia.org/wiki/Internet_refrigerator

> Now, 20 years later, we're seeing manufacturers building refrigerators that connect to the Internet. However, unlike the first smart refrigerator, these new devices are positioned to be a social media focal point for the household. Many don't provide any meaningful data about the contents of the refrigerator outside of the gadget-like ability to see a video image of the contents without opening the door, which could have been solved by installing a glass door.
>
> Suffice to say IOT enthusiasts like me scratch their heads at how something like this could possibly be useful much less sell well. Sadly, these new Internet refrigerators do indeed seem to be selling well, but I wonder if consumers have been sucked into the hype. For an interesting commentary on why the Internet refrigerator isn't for you, do a Google search and you'll find a lot of opinions – most negative (and yet, people still buy these things).[4]
>
> Let's judge the Internet refrigerator with my definition of IOT: does it enhance your life by providing you information about the world around you? Well, if you need to check to see how much milk you have while 3000 miles away from home, then I guess it may beneficial but for the multitude of us who prefer to just open the door and look before we go to the store, it may not be an IOT device.

IOT Services

Sadly, there are companies that tout having IOT products and services that are nothing more than marketing hype – much like what some companies have done by prepending 'cloud' or appending 'for the cloud' to the name. Fortunately, there are some good products and services being built especially for IOT. These range from data storage and hosting to specialized hardware.

Indeed, businesses are adding IOT services to their product offerings faster than anyone can keep up with the latest. And it isn't the usual suspects such as the Internet giants. I have seen IOT solutions and services being offered by Cisco, AT&T, HP, and countless startups and smaller businesses. I use the term IOT vendor to describe those businesses that provide services for IOT solutions.

You may be wondering what these services and products are and why one would consider using them. That is, what is an IOT service and why would you decide to buy it? The biggest reason you may decide to buy a service concerns cost and time to market.

If your developers do not have the resources or expertise and obtaining them will require more than the cost of the service, it may be more economical to purchase the service. However, you should also consider any additional software or hardware changes (sometimes called retooling) necessary in the decision. I once encountered a well-meaning and well-documented contracted service that permitted a product to go to market sooner than projected at a massive savings. Sadly, while the champions of that

[4] https://www.howtogeek.com/260896/why-buying-a-smart-fridge-is-a-dumb-idea/

contract won awards for technical achievement, they failed to consider the fact that the systems had to be retooled to use the new service. More specifically, it took longer to adopt the new service than it would have to write one from scratch. So instead of saving money, the organization spent nearly triple and were late to market. Clearly, one must consider all factors.

Similarly, if your time is short or you have hard deadlines to make your solution production ready, it may be quicker to purchase an IOT service rather than create or adapt your own. This may require spending a bit more, but in this case the motivation is time and not (necessarily) cost. Of course, it is a mixture of both cost and time.

So, what are some of the IOT services available? The following lists a few that have emerged in the last few years. It is likely more will be offered as IOT solutions and services mature.

- *Enterprise IOT Data Hosting and Presentation* – services that allow your users to develop enterprise IOT solutions from connecting to, managing, and customizing data presentation in a friendly form such as graphs, charts, etc. Example: Xively (`https://xively.com/`)
- *IOT Data Storage* – services that permit you to store your IOT data and get simple reports. Example: Sparkfun's IOT Data service (`https://data.sparkfun.com/`)
- *Networking* – services that provide networking and similar communication protocols or platforms for IOT. Most specialize in machine-to-machine (M2M) services Example: AT&T's cellular global SIM service (`business.att.com/enterprise/Family/mobility-services/internet-of-things`)
- *IOT Hardware Platforms* – vendors that permit you to rapidly develop and prototype IOT devices using a hardware platform and a host of supported modules and tools for building devices ranging from a simple component to a complete device. Example: Intel's IOT gateway development kits (`intel.com/content/www/us/en/embedded/solutions/iot-gateway/development-kits.html`)

Now that we know more about what IOT is, let's look at a few examples of IOT solutions to get a better idea of what IOT solutions can do and how they are employed.

A Brief Look at IOT Solutions

An IOT solution is simply a set of devices designed to produce, consume, or present data about some event or series of events or observations. This can include devices that generate data such as a sensor, devices that combine data to deduce something, devices or services designed to tabulate and store the data, and devices or systems designed to present the data. Any of these may be connected to the Internet.

IOT solutions may include several of these qualities whether they are combined into a single device such as a web camera, or into a sensor package and monitoring unit such as a weather station, or into a complex system of dedicated sensors, aggregators, data storage, and presentation such as complete home automation system. Figure 1-1 shows a futuristic picture of all devices everywhere connected to the Internet either to databases, data collectors or integrators, display services, and even other devices.

Figure 1-1. *The future of IOT – all devices, everywhere*[5]

Let's look at some example IOT solutions. The IOT solutions described in this section are a mix of solutions that should give you an idea of the ranges of sizes and complexities of IOT solutions. I also point out how some of these solutions leverage services from IOT vendors.

Sensor Networks

Sensor networks are one of the most common forms of IOT solutions. Simply stated, sensor networks allow you to observe the world around you and make sense of it. Sensor networks could take the form of a pond monitoring system that alerts you to water level, water purity (contamination), water temperature, detect predators, or even turn on features automatically such as lighting or feeding the fish in your garden pond.

[5] https://pixabay.com/en/network-iot-internet-of-things-782707/

If you, or someone you know, has spent any time in a medical facility, chances are a sensor network was employed to monitor body functions such as your body temperature, heart rate, respiratory capacity, or even movement range of your limbs. Modern automobiles also contain sensor networks dedicated to monitoring the engine, climate, and even in some cars road conditions. For example, the lane-warning feature uses sensors (typically a camera, microprocessor, and software) to detect when you drift too far toward lane or road demarcations. Manufacturing plants also employ sensor networks in monitoring and controlling the machines, conveyors, and more. Shipping clearinghouses also employ sensor networks to help route packages to the correct bins and ultimately to the correct trucks or planes for transport.

Thus, sensor networks employ one or more sensors that take measurements (observations) about an event or state and communicate that data to another component or node in the network, which is then presented in some form or another for analysis. Let's look at an example of an important medical IOT solution.

Medical Applications

Medical applications including health monitoring and fitness are gaining a lot of attention as consumer products. These solutions cover a wide range of capabilities such as the fitness features built into the new Apple Watch to Fitness bands that keep track of your workout and even medical applications that help you control life-threatening conditions. For example, there are solutions that can help you manage diabetes.

Diabetes is a disease that affects millions of people worldwide (`diabetes.org/`). There are several forms: the most serious being type 1 (`diabetes.org/diabetes-basics/type-1/?loc=db-slabnav`). Those afflicted with type 1 diabetes do not produce enough (or any) insulin due to genetic deficiencies, birth defects, or injuries to the pancreas. Insulin is a hormone the body uses to extract a simple sugar called glucose, which is created from sugars and starches, from blood for use in cells. Failure to monitor your blood sugar can result in dangerously low or high blood sugar levels, both of which can be life threatening and if not controlled can cause long-term damage to internal organs, nerves, and other areas. It is a most serious condition.

ATHLETES AND DIABETES

Professional athletes are some of the most physically fit people in the world. Many are examples of health and admired by fans and fellow athletes alike. In the past, if a professional athlete contracted a disease like diabetes type 1, their career would be over. Now, with modern medical technology, professional athletes are starting to overcome their condition and continue to compete.

One shining example is Ryan Reed, a NASCAR Xfinity stock car racer and driver of the number 16 Lily Diabetes Ford. In 2011, Reed was diagnosed with diabetes type 1 and told he would never race again. Since then, Reed has overcome his handicap through careful monitoring of his condition and has returned to racing.

> Not only has Reed returned to the sport he loves, he has won the season opening premier series race at Daytona International Raceway not once, but twice. Reed is proof that education, vigilance, and technology can make our lives better.

Type 1 diabetics must monitor their blood glucose to ensure they are using their medications (primarily insulin) properly and balanced with a healthy lifestyle and diet. If their blood glucose levels drop too low or too high, they can suffer from a host of symptoms. Worse, extremely low blood glucose levels are very dangerous and can be fatal.

One of the newest versions of a blood glucose tester consists of a small sensor that is left in the body for as much as a week along with a monitor that connects to the sensor via Bluetooth. You wear the monitor on your body (or keep it within 20 feet at all times). The solution is marketed by Dexcom (`dexcom.com/`) and is called a continuous glucose monitor (CGM) that permits the patient to share their data to others via their phone. Thus, the patient pairs their CGM with their phone and then shares the data over the Internet to others. This could be loved ones, those that help with their care, or even medical professionals.

Figure 1-2 shows an example of the Dexcom CGM monitor and sensor. The monitor is on the left and the sensor and transmitter are on the right. The sensor is the size of a small syringe needle and remains inserted in the body for up to a week.

Figure 1-2. *Dexcom Continuous Glucose Monitor with Sensor*

A feature called Dexcom Share permits the patient to make their data available to others via an app on their phone. That is, the patient's phone transmits data to the Dexcom cloud servers, which is then sent to anyone who has the Dexcom Share app and has been given permission to see the data. Figure 1-3 shows an example of the Dexcom Share CGM report from the Dexcom Share iOS app, which allows you to easily and quickly check the blood glucose of a friend or loved one.

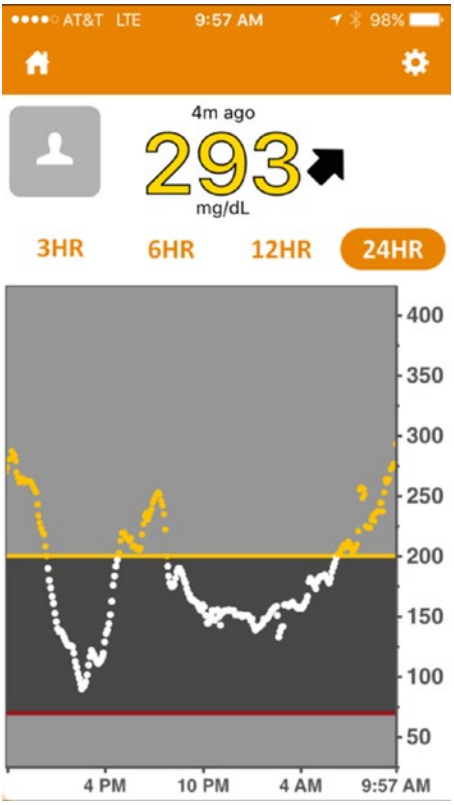


Figure 1-3. *Dexcom Share App Report*

Not only does the app allow the visualization of the data, it can also relay alerts for low or high blood glucose levels, which has profound implications for patients who suffer from additional ailments or complications from diabetes. For example, if the patient's blood glucose level drops while they are alone, incapacitated, or unable to get treatment, loved ones with the Dexcom Share app can respond by checking on the patient and potentially avoiding a critical diabetic event.

While this solution is a single sensor connected to the Internet via a proprietary application, it is an excellent example of a medical IOT device that can enhance the lives of not only the patient but everyone who cares for them.

Dexcom also provides a free Windows application called Dexcom Studio (`http://dexcom.com/dexcom-studio`) to allow patients to see the data their monitors collect and generate a host of reports they can use to see their glucose levels over time. Reports include averages, patterns, daily trends, and more. They can even share their data with their doctor. Figure 1-4 shows an example of the Dexcom Studio with typical data loaded.

Figure 1-4. *Dexcom Studio*

WHAT ABOUT BLOOD GLUCOSE TESTERS – GLUCOMETERS?

Until solutions like the Dexcom CGM came about, diabetics had to use a manual tester. Traditional blood glucose testers are single-use events that require the patient to prick their finger or arm and draw a small amount of blood onto a test strip. While this device has been used for many years, it is only recently that manufacturers have started making blood glucose testers with memory features and even connectivity to other devices such as laptops or phones. The ultimate evolution of these devices is a solution like Dexcom, which has become a medical IOT device that improves the quality of life for diabetics.

Combined with the programmable alerts, you and your loved ones can help manage the effects of diabetes. If you have a loved one who suffers with diabetes, a CGM is worth every penny for peace of mind alone. This is the true power of IOT materialized in a potentially lifesaving solution.

Automotive IOT Solutions

Another personal IOT solution is the use of Internet-connected automotive features. One of the oldest products is called OnStar (onstar.com) and is available on most late-model and new General Motors (GM) vehicles. While OnStar is a satellite-based service that has several levels and many fee-based options, it incorporates the Internet to permit communication with vehicle owners. Indeed, the newest GM vehicles come with a WiFi access point built into the car! Better still, there are some basic features that are free to GM owners that, in my opinion, are very valuable.

The free, basic features include regular maintenance reports sent to you via email and the ability to use an app on your phone to unlock, lock, remote start – all the features on your key fob remotely. This is a cool feature if you have ever locked your keys in your car! Figure 1-5 shows an example of the remote key fob app on iOS. Of course, there are even more features available for a fee including navigation, telephone, WiFi, and on-call support.

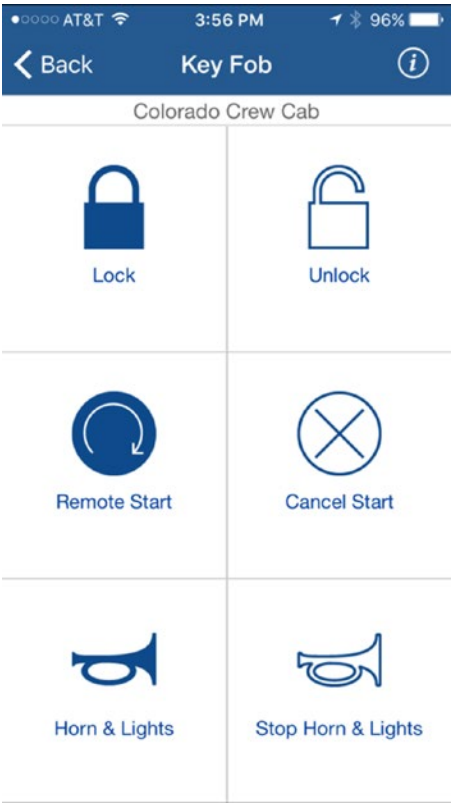


Figure 1-5. *OnStar App Key Fob Feature*

The OnStar app works by connecting to the OnStar services in the cloud, requesting the feature (e.g., unlock) that is sent to the vehicle via the OnStar satellite network. So, it is an excellent example of how IOT solutions use multiple communication protocols.

The feature I like most is the maintenance reports. You will receive an email with an overview of the maintenance status of your vehicle. The report includes such things as oil life, tire pressure, engine and transmission warnings, emissions, air bag, and more. Figure 1-6 shows an excerpt of a typical email you would receive.

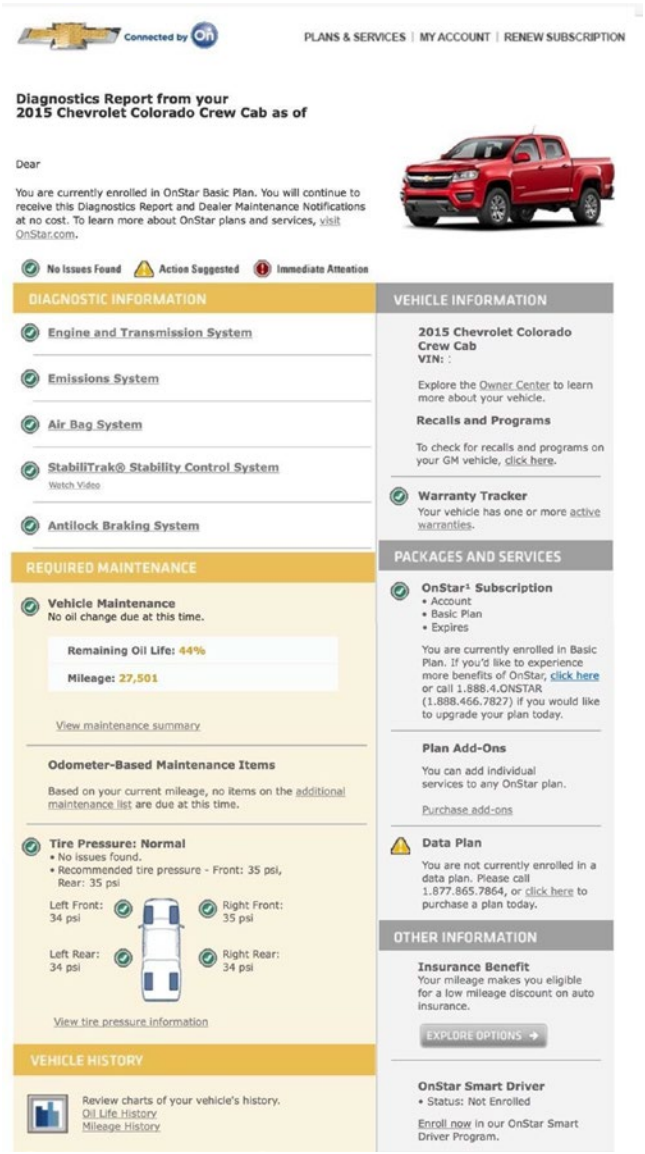


Figure 1-6. *OnStar Maintenance Report*