

The Internet as Material

Empowering the Next Phase of Connected Hardware Innovation



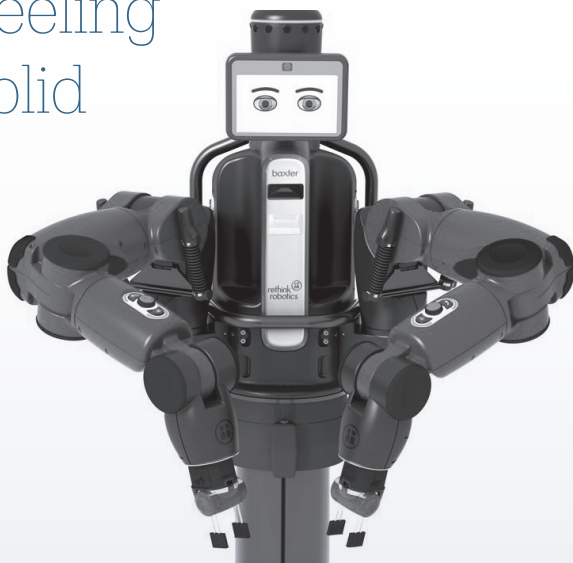
Ayah Bdeir

Solid

C O N F E R E N C E THE O'REILLY INTERNET OF THINGS CONFERENCE

“The future has a funny way of sneaking up on you. You don’t notice it until you’re soaking in it. That was the feeling at O’Reilly’s Solid Conference.”

—Wired



The traditional boundaries between hardware and software are falling. It's a perfect storm of opportunity for a software-enhanced, networked physical world. The new products and services created from the melding of software, hardware, and data are built by people who work across disciplines and industries. A vibrant new community is emerging, made up of business and industry leaders, software developers, hardware engineers, designers, investors, startup founders, academics, artists, and policy makers—many of whom have never come together before. They gather at Solid to be inspired, to make connections and launch conversations, and to plug into the future for a few days. Will you be a part of it?

Find out more at **solidcon.com**

The Internet as Material

*Empowering the Next Phase of
Connected Hardware Innovation*

Ayah Bdeir

The Internet as Material

by Ayah Bdeir

Copyright © 2015 O'Reilly Media, Inc. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://safaribooksonline.com>). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

Adapted by: Troy Mott

Editor: Brian Sawyer

Production Editor: Kristen Brown

Interior Designer: David Futato

Cover Designer: Ellie Volckhausen

Illustrator: Rebecca Demarest

April 2015: First Edition

Revision History for the First Edition

2015-04-30: First Release

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *The Internet as Material*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

While the publisher and the authors have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the authors disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-491-92635-2

[LSI]

Table of Contents

The Internet as Material: Empowering the Next Phase of Connected Hardware Innovation.	1
Internet as Material	2
Democratizing Hardware	3
littleBits Library	7
Audience of One	11
Don't Reinvent the Wheel	12
Dumb Sensors, Terminals, and Displays	12
Internet as Building Block	13
Solving a Problem for One Person is Beautiful	15

The Internet as Material: Empowering the Next Phase of Connected Hardware Innovation

*The following document is adapted from the **keynote address** by Ayah Bdeir, on **The Internet as Material: Empowering the Next Phase of Connected Hardware Innovation**, given at the Solid 2014 conference. Ayah opened her Solid keynote with praise for Neil Gershenfeld, who **delivered a keynote** earlier in the same day on the third digital revolution.*

Highlights

- Take advantage of the Internet as Material
- Democratize hardware to make it more accessible
- Discover the power of hardware modularity
- Create for an audience of one
- Tap into the Internet as a building block

Before I get started, I'd like to vouch for something that was said in an earlier talk about Neil's **How to Make Almost Anything** class. For me, this was the hardest thing I've ever had to get into. It was harder than getting into MIT. But to be honest, that class by Neil Gershenfeld changed my life.

Internet as Material

My talk here today is about the Internet as Material. We all know that the Internet of Things is a field that comes with a lot of promise to make our lives easier. The promise to get our thermostats talking to our fridges, thus preventing us from forgetting to buy milk, but also the promise of making our relationships with technology even better.

There have been many technologies and waves of technologies that have also come with a lot of promise. In the 40s, the 50s, and the 60s, when the first washing machines and dishwashers came about, it was all about the promise of how technology was going to make our lives easier.

And yet, when we look at it, most devices are closed, finished products that have prescribed and predefined functionality. These devices make a guess at how to make our lives easier. A few days ago, I was reading a Pew study about the Internet of Things, and this quote by Harry Rheingold really caught my eye: “We will live in a world where many things won’t work and nobody will know how to fix them.”

When I read this I had two thoughts. The first thought, unfortunately, is that we already live in a world where many things don’t work and we don’t know how to fix them. In fact, when Hurricane Sandy happened (and I’m based in New York) a friend with a master’s degree in computer science was asking me why, after the hurricane had passed, the lower part of Manhattan didn’t have power and yet the upper part of Manhattan did have power. There were these fundamental misunderstandings about how the electricity grid works and how power stations work that suggest perhaps we have reached a point where technology has become so far from us that we don’t really understand it.

The second thought I had about Harry’s quote is that perhaps this is happening due to the way that technologies are parachuted onto us. The TV industry is an example of a technology that came with a lot of promise, and yet, when I go to buy a TV, I have no idea about the difference between 3D, 4D, and 4K ([Figure 1-1](#)). I have no idea what works. I just want something to display pictures.

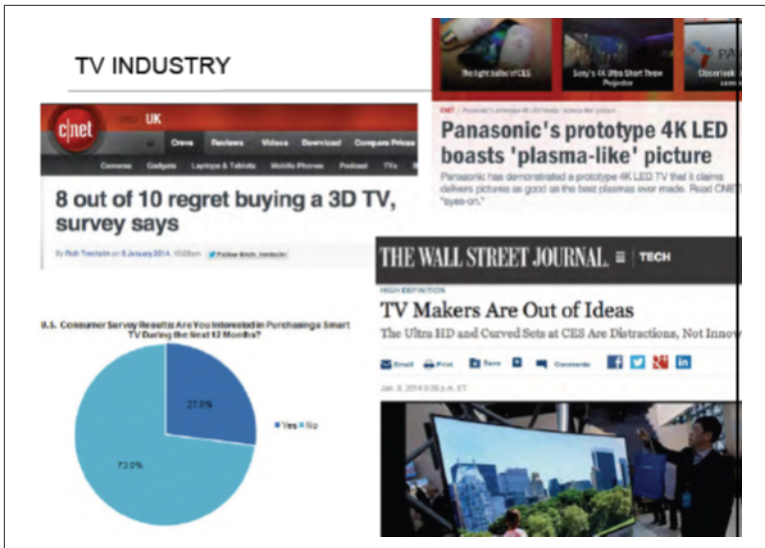


Figure 1-1. Which TV should I choose?

But it doesn't have to be this way. I am optimistic that we can change this, that we can make hardware and the Internet of Things work in our favor and do what we want them to do to really improve our lives.

Democratizing Hardware

Some of society's most transformative technologies have started in the hands of experts. And then something or someone came along and democratized them, making them accessible to everyone. And we've really had a chance to transform society.

A few years ago, I started working on this problem with a particular focus on hardware. So, how do we democratize hardware? For me, there are four principles:

1. Lowering the barrier to understanding
2. Lowering the barrier to iterating
3. Making it universal
4. Raising the ceiling of complexity

If you look at the left side of Figure 1-2, you have a raw proto circuit, or a breadboard circuit that is trying to have a function. Nobody wants that; nobody is trying to make that. That is a means to an end.

On the right side of **Figure 1-2**, you have a device that is finished and polished and ready for you to use, but it is also closed, and it doesn't have a lot for you to integrate.

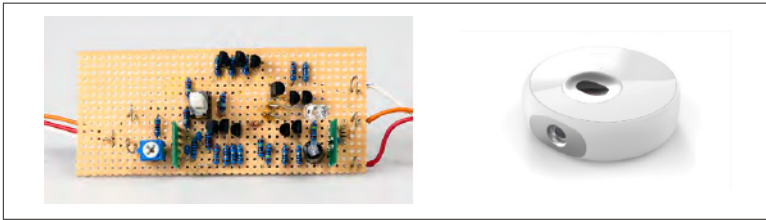


Figure 1-2. A breadboard proto circuit (on the left) and a closed, finished, and polished device (on the right)

If I compare this to the world of software, on the left side of the figure, the proto circuit feels to me like assembly code. It feels to me like the low-level version of hardware technology. And the right side feels like an app: something that is finished, that you consume and interact with.

But what is in the middle? What is the modular approach to hardware that we can learn from software? What is the C++ for hardware? What is the object-oriented version of hardware? My approach to this question was to make hardware modular. In early 2008, I started working on this problem. I actually went to purchase a modular circuit and I couldn't find anything, so I started working on it myself. I started making circuits that were modular, that each had a single function. It was a delicate balance between creating enough functionality in the hardware so that it is meaningful, but also making it small enough so that you don't have to constrain what people are trying to do with it.

Figure 1-3 shows a cardboard circuit with copper tape on it, which is functioning as a sound recorder.

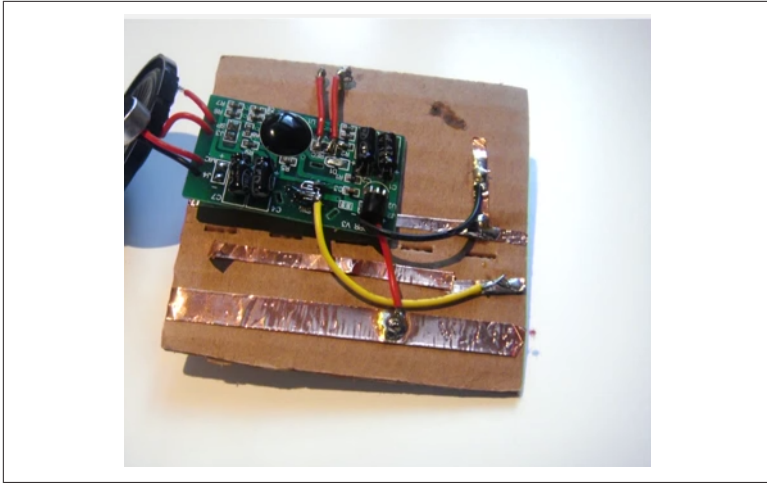


Figure 1-3. An early sound recorder module

The circuit shown in **Figure 1-4** is a battery. The idea here is that, when creating something that is battery powered, you don't have to reinvent how a battery distributes power.



Figure 1-4. An early battery module

I then started to experiment in different ways. This took many years (it was early 2008) to really think about how to create a system that was modular, but that was also easy to use and iterate with.

Figure 1-5 shows a later creation, which uses magnets to snap modules together, thus creating a simple device. In this case, I have created a dimmer with an LED and a pressure sensor.

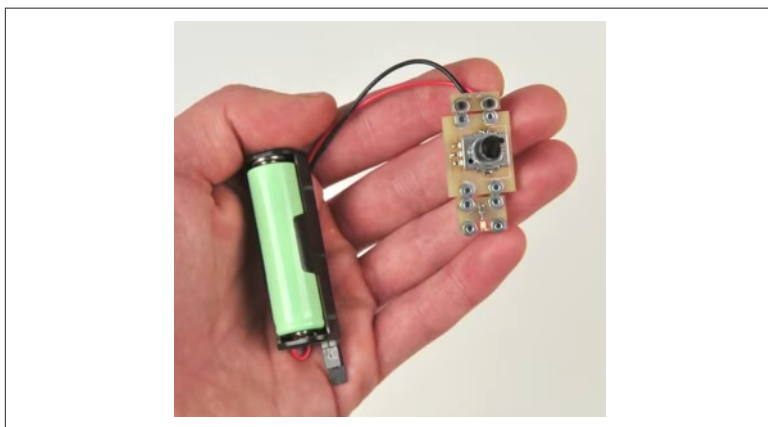


Figure 1-5. Modules are connected with magnets

About 28 or 29 prototypes (3.5 years later), the littleBits library looks like Figure 1-6.



Figure 1-6. The littleBits module library, which continues to grow

These electronic modules each have some very simple and some very complex functions, but they all work within the same system to make their iteration with hardware very easy. The goal here is to put the power of electronics into everyone's hands.

I won't go through all of the details, but the left side of [Figure 1-7](#) shows a raw circuit that is breadboarded with wires and hundreds of components, with a finished product on the right.

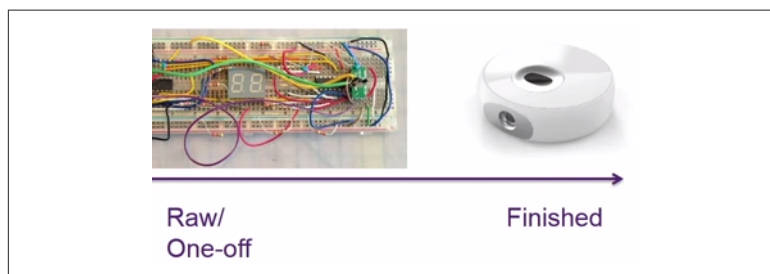


Figure 1-7. A raw circuit (on the left) contains hundreds of components, and the finished device (on the right) contains the same components

What if we inject something in between? In this case, we have a seven-segment display that shows a number. You want to see a number. You don't care about how the circuit is made, especially if you aren't an engineer. For me, the device shown on the right side of [Figure 1-8](#) is the middle ground, or the C++ version of it.

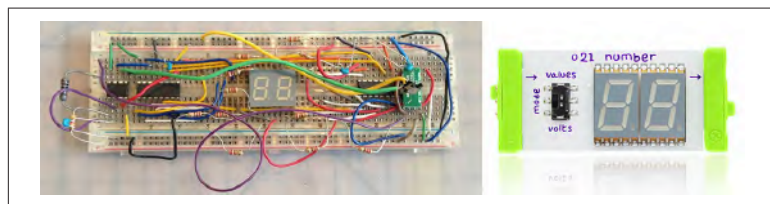


Figure 1-8. The raw circuit (on the left) and the middle-ground product (on the right)

littleBits Library

The concept is simple, but the execution is actually pretty difficult. For you to think ahead of time about all of the possible circuits that you might come up with is actually a very complex task. And manufacturing is not an easy thing to figure out, but it is a worthwhile challenge, because it empowers people in a way that we've never imagined possible.

Today, the littleBits library contains close to 60 modules, and there are another 40 on the way. This is a really extensive and powerful,

but also easy-to-use, modular hardware library. It enables people with absolutely zero experience to create highly complex behaviors with electronics.

There are billions of different combinations of circuits that you can make. This provides you with freedom. You don't have to hire an engineer or know exactly what outcome you want as you set out to make a project. Yet this still allows for complexity, since raising the ceiling of complexity is important. Like Neil Gershenfeld mentioned here at the conference, these modules are not toys. They are tools to empower people.

Figure 1-9 shows two examples of wireless communication devices that allow two things to communicate wirelessly. On the left side is a prototype that takes hundreds of components and dollars to build, and there is a lot of room for error. Or instead, you can use modules (shown on the right side).

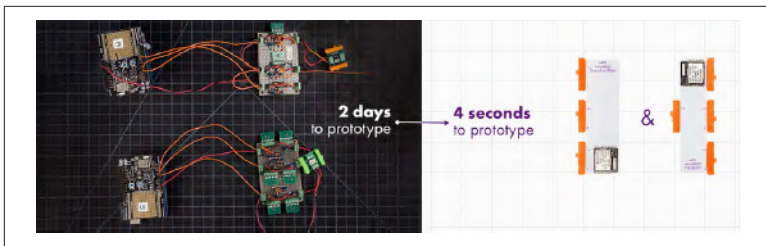


Figure 1-9. A prototype with hundreds of components for wireless communication (on the left) and the same features in a finished product (on the right)

Let's look at some example projects that use modules. Figure 1-10 shows a mounting system (built by an industrial designer) for an iPhone that allows you to take video in an easy and consistent way.



Figure 1-10. This mounting system lets you record video from a phone in an easy fashion

Figure 1-11 shows a slow water cooler that we have in our office. We don't like to wait for it to fill our glasses, so there is a timer that stops dispensing water when it is ready.



Figure 1-11. No need to wait around with the help of this water cooler timer

Figure 1-12 shows a wireless light fixture. It allows you to dim and rotate light in different ways. Again, this is made by a designer who had no experience in electronics.

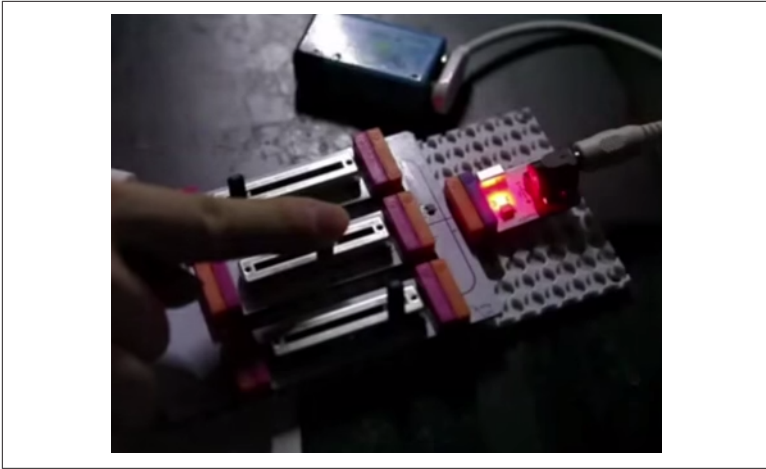


Figure 1-12. This device allows you to dim and rotate light as needed

Figure 1-13 shows a robot delivering a screwdriver across the office. It is wirelessly controlled, and we call it the FaceTime robot.

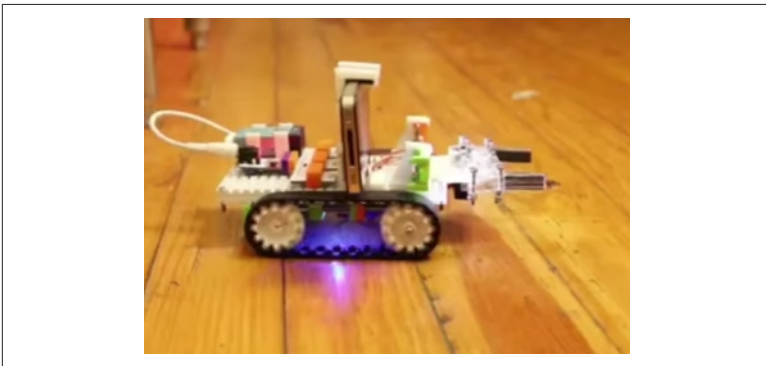


Figure 1-13. Our FaceTime robot runs mini errands for us

We also have our version of **Bot & Dolly** that picks up a component and then puts it back.

Audience of One

These examples took anywhere from minutes to hours to create, but we aren't talking about days. And we aren't talking about experience, because sometimes you want to do something simple. One of our team members, Andrew, is pretty lazy. He doesn't want to go to the bathroom only to discover that it is occupied. So, he made this little device (shown in [Figure 1-14](#)) that displays whether or not the bathroom is in use, without having to go all the way there.



Figure 1-14. A visual cue lets you know when the bathroom is available

This is a funny project, but it illustrates an idea that I think is very important. Part of the problem with having technology (particularly hardware) solely in the hands of experts in companies is that they are going to guess what your needs are.

And there has to be a certain critical mass of these needs for a project to warrant existence. Sometimes, you just need to know whether someone is in the bathroom. Maybe you are the only person in the world with that need, but that doesn't mean that the product shouldn't exist. It may not warrant having mass manufacturing, because it's a product with an audience of one. But that doesn't mean it is any less important than any other things that you might use.

For one of our community members, we made this huge device ([Figure 1-15](#)) with LEGO and littleBits sensors to be a mechanical musical instrument.

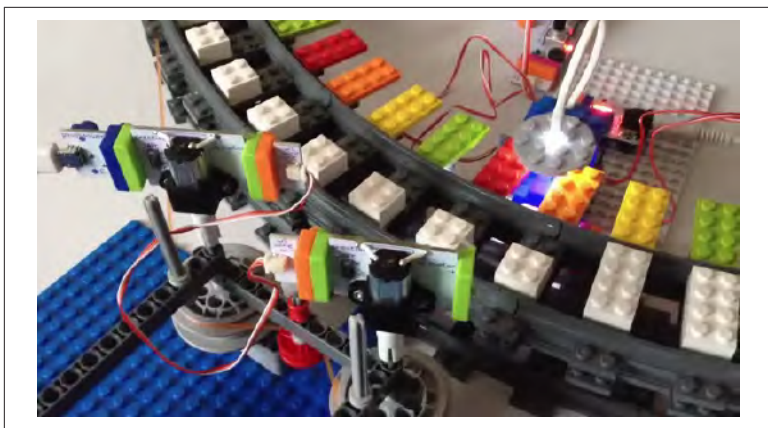


Figure 1-15. LEGO and littleBits team up to work in a mechanical instrument

Don't Reinvent the Wheel

The lines continue to blur between art and technology. We want to make tools available for people, without necessarily having a function or startup in mind. Sometimes, you want to make art without acquiring funding. And this art needs to use the technology of the day—in this case, electronics.

We are also enabling more makers, designers, and prototypers to create more things. Even those with extensive experience in hardware electronics can cut down on their prototyping time when they don't have to reinvent the wheel so often. On the other side of the spectrum, people with no experience at all can use modules to make powerful, modern, and computational things.

Dumb Sensors, Terminals, and Displays

Going back to the Internet of Things and how that is relevant, I see people putting a lot of devices out there that are finished products. These things are prescribed solutions to solve a problem. But I also see a problem of overkill. A lot of these devices are individual computers. Why should a temperature sensor that you are using at home be a computer? Why should a switch that tells you whether your door is open or not be a computer? Why are we adding so much computational power, when all we need is computation? It's not necessary to put a computer on these peripherals. Sometimes, all we

want are dumb sensors, and dumb terminals, and dumb displays that then plug into something we've already built, which is the Web.

We've already built the Internet. Are we going to rebuild it now in the physical world? I believe that we don't have to, that the idea is to create these entry points to the Web. And so I go back to an idea that Neil wrote about in *Scientific American* (with Raffi Krikorian and Danny Cohen) in 2004, at the very beginning of his field. My first interaction with the Internet of Things was learning about **Internet-0 from Neil's article**. It explores the idea of a distributed open network of sensors and terminals, and sometimes dumb devices that aren't computers that enable you to do different things. So the idea comes back as: "can we make the Internet the building block?" Can it become a building block that empowers people to invent with the Internet in the same way that you would invent with light, sound, cardboard, or paper, and really make it a material?

Internet as Building Block

I leave you with one final example. In **Figure 1-16**, the Internet is the cloud in the middle and you are creating an automated fish feeder.



Figure 1-16. littleBits gives your crafted device Internet access

You swap out the timeout module, and put in an MP3 player with a speaker. Send it a text message, and you've now created your own wireless device for disseminating music (**Figure 1-17**).



Figure 1-17. littleBits allows you to easily snap in an MP3 player

Swap out the MP3 player, and instead add a servo. Send an email to it, and now you've created your own Nest (Figure 1-18). An old-school Nest, but still.



Figure 1-18. Create your own old-school Nest product

And then you put one in Tokyo and one in New York, and you embed it with other materials, such as, in this case, a plush toy.

We now have Collin telling his daughter he loves her from across the world (Figure 1-19).



Figure 1-19. Connecting Tokyo and New York in a project with cloud modules

Solving a Problem for One Person is Beautiful

We want to enable people to create solutions that can solve a problem for one person. They are solutions for people that won't have a manufactured product created for them. And that is okay, and that is beautiful, and that is essential. And that is part of redemocratizing hardware and democratizing the Internet of Things, so that it's no longer like the TV industry, where you are sitting and consuming, hoping for the next Best Buy salesperson to help you. We are looking for partners, for people with ideas and feedback who want to work with us.

About the Author

Ayah Bdeir is the founder and CEO of littleBits, an award-winning library of electronics dubbed “LEGOs for the iPad generation.” Bdeir is an engineer, interactive artist, and one of the leaders of the open source hardware movement. Bdeir’s career and education have centered on advancing open source hardware to make education and innovation more accessible to people around the world. She is a co-founder of the Open Hardware Summit, a TED Senior Fellow, and an alumna of the MIT Media Lab. Bdeir was named one of *Inc. Magazine*’s 35 Under 35, one of Fast Company’s 100 Most Creative People in Business, and one of *Popular Mechanics*’ 25 Makers Who Are Reinventing the American Dream. littleBits was named as one of CNN’s top 10 Emerging Startups to watch. Originally from Lebanon and Canada, Ayah currently lives in New York City.
