Arjmand Samuel *Microsoft Research* **Dean Mohamedally** *University College London*
Nilanjan Banerjee *University of Maryland, Baltimore County* **A.J. Brush** *Microsoft Research* **Ratul Mahajan** *Microsoft Research*

Editors: Nilanjan Banerjee and Sami Rollins

LAB OF THINGS IN

EDUCATION

Photo, (this page) istockphoto.com

Lately, much has been said about the Internet of Things and how it is going to change how we will live, work and play. Strategy Analytics forecasts that by 2020, every person on the planet will carry four connected devices [1]. The majority of this growth will be driven by the interconnection of devices, sensors, smart objects and the like, and is expected to usher in changes to almost all aspects of our lives. This change will require, on the one hand, large-scale design, development and deployment of cloud and network systems; and on the other hand, design of hardware sensors, actuators, software middleware, and network protocols.

The Internet of Things (IoT) will require global-scale data centres to store and serve IoT data generated by these billions of devices in a scalable and power-efficient manner. The design of the IoT network will rest upon innovations in low power, reliable and efficient network systems. The software abstractions enabling interconnection of IoT will implement interoperable APIs providing access to device functionality and data. Sensors and devices will run lightweight operating systems and operate reliably in resource-constrained environments. These sensors and devices will be embedded in wearables, cars and homes, with unique form factors and operating constraints.

This sea of change requires engineers, programmers, designers, developers, and researchers who can not only design and develop the infrastructure of the IoT, but also can envision and create use cases and scenarios that allow seamless integration of IoT in our lives. Introducing the concepts of IoT early in engineering and other fields of study, such as design, management and operations is the only way to prepare today's students to be effective IoT developers and designers of tomorrow. In this article we argue that a hands-on approach to IoT education and teaching can result in the workforce of the future being better equipped to understand, and subsequently develop and deploy IoT technologies, solving real-world problems.

Microsoft Research Lab of Things (LoT) was released in July 2013 as a research platform for using connected devices in homes and beyond. LoT allows a large number of disparate devices to be connected together and controlled by applications, called experiments. The LoT platform allows these experiments to be scaled up in a number of locations (such as multiple homes) by providing remote monitoring and updates. LoT also allows data from experiments to be collected, both locally, and in a variety of cloud storage systems. The LoT platform is available free of charge for academic and teaching purposes in source code form. Contributions to the LoT platform are encouraged in the form of drivers and applications allowing the community to benefit from each other's engineering efforts.

Since its release, in addition to being

used for a number of research studies, we are also seeing an active interest in using the LoT platform for teaching and student projects. To the best of our knowledge, LoT is the first community-based platform, allowing interconnection of devices, execution of experiments, and collection of data for research and teaching. LoT allows experiments to scale up and to be deployed in different locations by providing greater visibility and control over deployments through its management services. In this article, we describe the design of the LoT as a platform for teaching and demonstrate its success in an IoT course at the University of Maryland as well as in student projects at University College London and University of Maryland. This article can serve as a baseline for instructors who want to teach IoT classes and students who want to rapidly prototype IoT applications. We next provide an overview of the Lab of Things Platform and its key design principles.

THE LAB OF THINGS ARCHITECTURE

Before we describe how the design principles of Lab of Things can be applied to teaching, we first briefly discuss the architecture of LoT. The design of the LoT infrastructure can be broken down into a client-side platform deployed in each location where sensors and devices are deployed; and a set of cloud services that enable Lab of Things to scale, share data and manage deployments. Figure 1 shows the overall architecture of the Lab of Things. HomeOS [3] forms the core of the client side component, which is deployed on a Windows 8.x machine, called the HomeHub. An outline of the three functions of HomeHub are as follows: (1) ***Protocol Abstraction*** that allows devices communicating over multiple protocols to be addressed and used transparently; (2) ***Experiment Execution*** that allows logic of the experiment (also called an app) to be executed on each HomeHub; and (3) ***Data Aggregation*** that allows data collected as part of experiments to be aggregated and stored locally and in cloud systems of choice such as Windows Azure. The user interface of each app is implemented using HTML5.

The Lab of Things cloud-side infrastructure provides the following services. (1) A ***Management Portal*** aggregates deployment-wide data from all deployed HomeHubs and presents it in the form of a webpage. Vital details for each HomeHub, such as memory and CPU usage, are also available. (2) An ***Update Service*** allows software update of all client side components across all deployed HomeHubs. It can also be used to update configuration and drivers, allowing researchers to move from one experiment to the next without the need to physically retrieve, re-install and re-deploy HomeHubs. The Update service is controlled using a client-side interface, which makes scaling Lab of Things easy by reducing the effort required for software updates. (3) An ***Alert Service*** allows definition of email alerts to report disruptive events such as the HomeHub going offline. (4) A ***Data Collection Service*** allows each HomeHub to store data in major cloud storage systems such as Microsoft Azure and Amazon S3. It also allows storage of experimental data on the HomeHub or on any networked server. Data stored in the cloud storage system of choice can then be analyzed using standard tools. In the next section, we describe how some of the design principles of Lab of Things are tuned to classroom teaching and student projects.

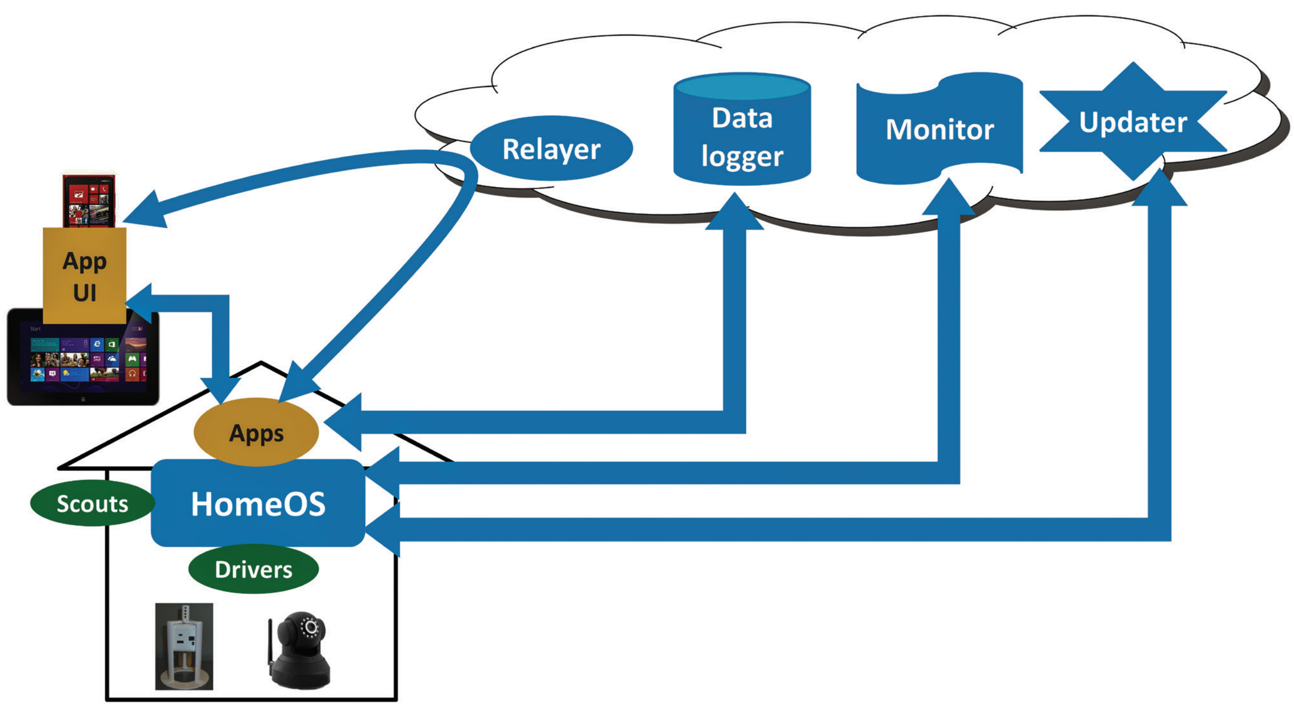

FIGURE 1. Lab of Things architecture: The LoT SDK provides all elements in blue. Elements in green are required by each device and written and shared by members of the community. Elements in yellow are written by the researcher for their experimental deployment.

LAB OF THINGS AS A PLATFORM FOR TEACHING

Lab of Things was built on several design principles that make it applicable to teaching and developing class projects. Some of these design principles are described below.

Support a wide range of custom and off-the-shelf devices: LoT supports most popular protocols, including Z-wave, ZigBee, Wi-Fi and Bluetooth. In order to support projects where sensors and devices are prototyped, LoT also supports Arduino and .NET Gadgeteer [2]. This design principle allows a wide range of experimentation by students from various backgrounds and levels, from undergraduates to graduates. As we demonstrate through the student projects in Section 5, the versatility also allows rapid prototyping of fairly complex projects in a short time.

Allow scale and diversity of deployments: Being able to easily scale up an experiment beyond a lab setting is a learning experience for students. Moreover, being able to deploy in different geographies and still being able to control experiments allows students to experiment with ideas and scenarios involving rich and meaningful data. LoT provides support for cloud services, built using Windows Azure that can be used to run large-scale experiments across multiple homes.

Enable the community to share code, experiments and data: An important aspect of the Lab of Things is the participation of the community of teachers, researchers and students to grow the platform and contribute code. This allows students to re-use ideas and software components already developed by their peers. The repository of drivers and applications of LoT has grown exponentially and the code-base now supports devices like the Mindwave EEG headset. The community also allows contribution of material such as courseware and slides (e.g., the Smart Home Automation course taught at University of Maryland, Baltimore County).

Over the past two years, LoT has been adopted in several classroom projects and has proven effective in a variety of application scenarios. The SoundChoice [4] project, for example, leverages the platform for ***device prototyping.*** The project involves the development of a wearable, watch-like device for people with hearing impairments. LoT provides the developers with connectivity and an application execution environment, allowing them to concentrate on the design and validation of the sensor device itself. LoT has also been popular for projects that wish to easily interconnect devices working on ***multiple protocols,*** for example, capturing events from one type of device (e.g., a Z-wave-based door sensor) to trigger another device (e.g., a Wi-Fi camera that may capture an image). Finally, LoT has also been used for projects focused on ***data collection and visualization*** storing data in the cloud and performing visualization or analysis using a variety of tools.

FIGURE 2. Snapshots from the Demo Session for the Smart Home Automation Course.

LAB OF THINGS IN A SMART HOME AUTOMATION COURSE (UMBC)

Lab of Things is a software framework for rapid prototyping of Internet of Things platforms and devices. This section describes such an exploratory course – *Systems for Smart Home Automation* – that was taught as an elective in the Computer Science and Electrical Engineering department at University of Maryland, Baltimore County by Dr. Nilanjan Banerjee. The goal of the course was to provide hands-on experience to students on programming sensors and devices using an existing IoT framework. Below, we describe why we used LoT as the platform for teaching the course; the different modules that the course covered; the challenges that we faced with using the platform; and illustrate several sample projects.

Why Lab of Things for an Internet of Things course

IoT is a complex, inter-disciplinary area of research. It involves development of sensors, networking protocols for communication between sensors, data capture and analytics, and the use of cloud services and mobile systems. Moreover, the complex interactions between these elements make understanding concepts in IoT systems difficult. For instance, consider the following IoT system: a network of occupancy sensors deployed in a home that continuously collects data using a Zigbee network, runs analytics on the data to determine room-level occupancy, and controls a thermostat. Understanding how the system works requires understanding of the sensor, the Zigbee networking protocol, as well as the data collection and analysis algorithms governing the underlying control. Designing a course that incorporates these varied Computer Science and Computer Engineering concepts and provides an understanding of the interaction between different subsystems is non-trivial and challenging. LoT, however, provides a framework that abstracts and hides the complexity of the underlying distributed hardware and the protocols for communication between the hardware elements. The idea behind LoT is similar to an Operating System, a layer of software that abstracts the complexity of the underlying hardware from the applications. However, like an Operating System, LoT also allows an ambitious developer to implement drivers for a new hardware (sensors) or write applications assuming that drivers exist. Like an Operating System, LoT exposes common abstractions (called ports) that allow applications to interact with sensors with minimal knowledge of the operation of the underlying hardware. Such an architecture helps an instructor teach basic design concepts governing Internet of Things without teaching complex sensing and network issues. The Smart Home Automation course at UMBC uses LoT as a basic framework for understanding IoT concepts applied to smart home automation. Below we describe the design principles that form the foundation of the course.

Design principles governing the Home Automation course

The Smart Home Automation course was designed using two key design principles (1) teach complex home automation concepts in a hierarchical fashion; (2) provide hands-on, practical experience to students. Below we describe, using examples, how each design principle was implemented.

Principle 1: Teach Complex Home Automation Concepts Hierarchically. Given the complexity of home automation and IoT systems, the course was divided into three hierarchical modules. In the first module, the course material covered components explaining high-level LoT concepts for creating home automation systems. The students were taught how to implement applications using the LoT platform and quickly prototype home automation systems. The key idea was to ensure that students learned to develop sophisticated IoT applications without in-depth understanding of the underlying sensors and communication protocols. For instance, the students during class lab sessions were able to design a simple occupancy-based light control system. The system combined a Z-wave switch to control a lamp and a Z-wave motion sensor that collected data on user movement. The Z-wave drivers are provided in the LoT distribution, and the students only had to write an application that captures data from the motion sensor and controls the binary switch. In the second module of the course, the students were taught how sensors and cameras are used in home automation work. Specifically, the students were introduced to the Z-wave networking protocol and the variety of Z-wave home automation sensors available commercially. Next, they were taught how to program the devices as well as how to write drivers using the LoT platform. Another key component of the second module in the course was learning about the Kinect [5] and Z-wave cameras [6]. Students learned how to use the Kinect SDK in conjunction with the LoT for depth imaging, skeletal imaging, and gesture recognition. The second module of the course was designed such that students with previous LoT experience were able to delve deeper to understand the underlying protocols for operating the sensors and implement drivers, while students with less experience could focus on building high-level applications without in- depth understanding of the underlying hardware. The third module of the course focused on *remote control,* i.e., the ability to perform smart home automation tasks when the user is not physically present at home. It included an introduction to web and cloud services, as well as design of rudimentary smartphone applications for control. The three modules combined gave students a comprehensive view of Internet of Things systems applied to home automation.

Principle 2: Providing hands-on experience through class demonstrations and lab sessions. Hands-on experience with building sensor-based systems is critical to developing an in-depth understanding of the design and realization of IoT systems. To this end, the course was taught in a lab format. Every class session included a short lecture followed by a task that students performed as a group. The tasks involved building a small software module to understand a concept taught in class. For instance, after the lecture that introduced the depth-imaging feature of the Kinect, students were required to build an application that would color-code an image to visualize the distance of an individual from the Kinect. The students had to solve the problem in groups of four and demonstrate a working prototype by the end of the class. The format not only gave students experience working in groups, it also provided hands-on experience with building software specifically for sensor and camera systems. The class also included a mandatory group project where students worked in groups of two over the course of two months to build and demonstrate a real home–automation application using LoT. The applications were showcased in a poster and demonstration session at the end of the semester that was open to faculty members and also to local entrepreneurs. Figure 2 shows picture taken during the event. The applications showcased included voice-controlled music systems, a SmartWall application, a mind-controlled home automation system and several others.

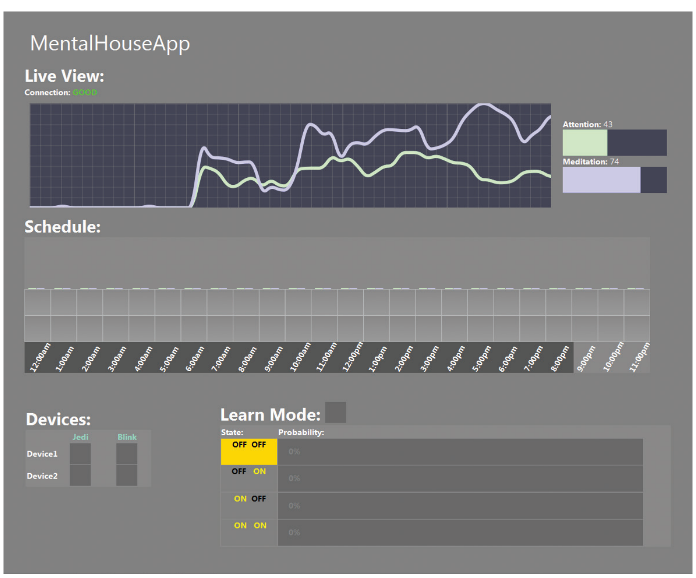

FIGURE 3. The user interface of the MentalHouse application – EEG headset driven home automation system.

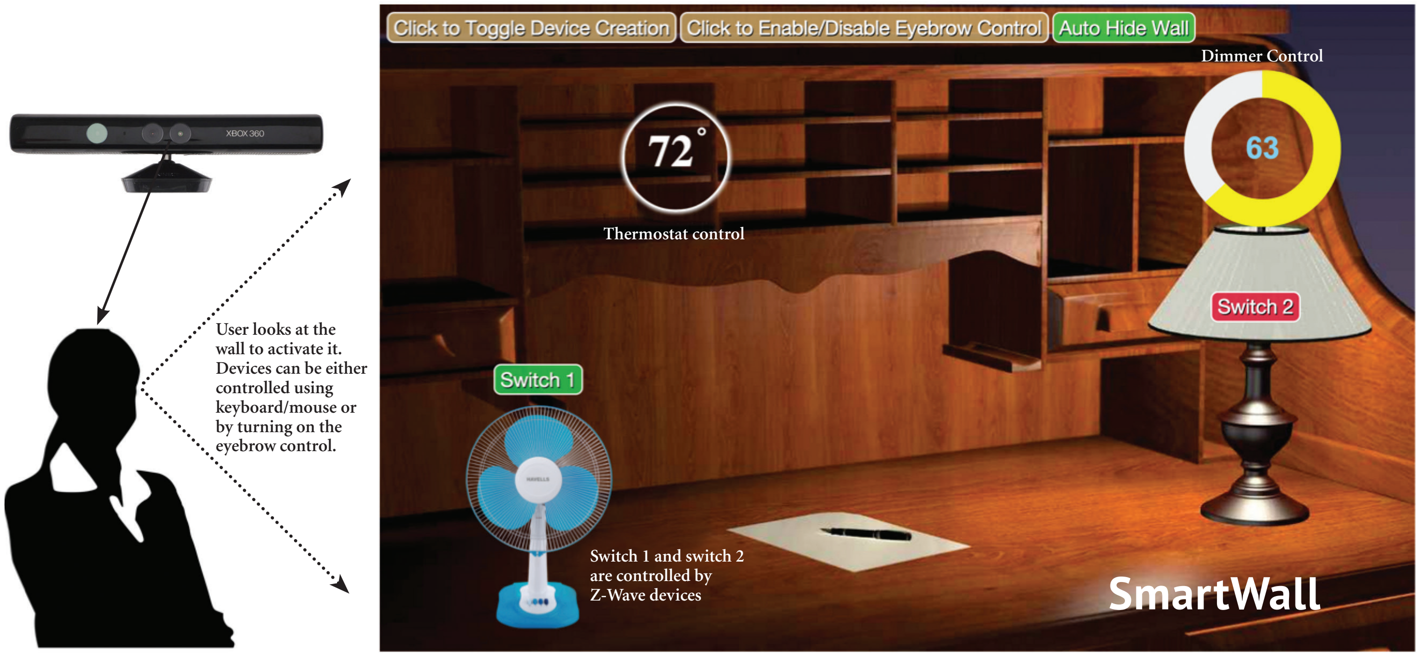

FIGURE 4. SmartWall interface. The system allows exporting your laptop screen to a wall and controlling appliances through controls on the SmartWall.

EXAMPLE STUDENT PROJECTS

We next describe in detail three example student projects that illustrate how LoT enables complex projects to be designed and built by students in a short period of time.

MentalHouse

(By Nickolas Kostriken, UMBC)

The idea behind this project was to create a set-up that would allow a home user to feel like they had psychic power over their home (Figure 3 illustrates the user-interface for the prototype). This concept was extended into two modes, the first of which essentially behaves like a brain-enabled remote control for connected appliances, and the second one using neurofeedback to optimize the user's comfort level by stochastically powering connected devices on and off. The first mode, nicknamed Jedi Mode, would be ideal if you wanted to have direct control over specific devices in your home. Just by closing your eyes and clearing your mind, you can turn them on and off. This also works with blinking. For instance, *just blink three times to toggle the power to another set of devices.* The second mode, called Learning Mode, would be ideal if you wanted to maintain a certain mental state without having to keep track of which devices you have left on or off. For example, if you wanted to be relaxed, this mode would adjust the environment's variables to maximize your level of relaxation, perhaps by turning off all the lights and turning on a fan.

This project was built using an off-the-shelf EEG recording device called the Mindwave Mobile (by NeuroSky) [7]. It is an inexpensive, battery-powered headset that uses a single dry electrode to collect brainwave data from the user's forehead and wirelessly stream it back to a computer over a Bluetooth connection. Using Lab of Things, we were able to quickly create a driver for the Mindwave that handled the connection, parsed the incoming data, and exposed a set of roles that we could use to read the live stream of values coming from the device. We were also able to connect to and control Z-Wave enabled smart switches using the Z-Wave driver provided by LoT. The app's front end was written in HTML and Javascript and it utilizes some of the powerful resources available to modern web browsers, like SVG rendering for charts and graphs.

SmartWall

(Piyush Waranpande, UMBC)

SmartWall, a project built by a single student, enables the following scenario: *While sitting on your LoT-connected computer, you look to your right where the air conditioner is placed. Just by looking to the right, you activate the SmartWall to your right side. Without disrupting your work, you drag your mouse to the A/C control switch presented by SmartWall. You adjust the temperature using clicks and scrolls and look at your computer screen to continue your work. You realize that the lamp on your left is not needed. You look to your left and blink your eyes looking at your lamp. The lamp is turned off. You look back at your screen and resume your work.* These actions are performed with minimum extra effort.

In SmartWall, the Microsoft Kinect tracks the movement of a person's head, and controls that wall that is activated. Once activated, the control is passed to the LoT application, which detects various Z-wave sensors and switches associated with the SmartWall. The SmartWall provides controls for devices such as dimmers and switches. It also displays information provided by the sensors. Finally, the Kinect detects blink gestures and controls appliances.

The project was implemented using the Microsoft LoT platform. Excluding the time required to learn the functioning of the platform, the project took approximately two weeks to be completely functional in its first phase. Most of the code was readily available, including code for controlling Z-wave devices and sensors and for controlling the Kinect. The only code to be written was wrapper classes that integrate the different elements together and the associated code required to implement a usable web-based interface for the application. SmartWall is an example of a project where a single student, in a short period of time, was able to master LoT concepts and build a working prototype of a system that uses multiple sensors.

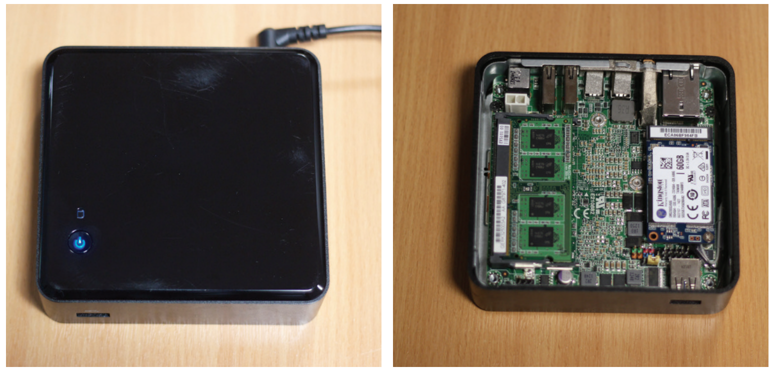

FIGURE 5. Captain Device Prototype

Captain Device *(UCL)*

A third project that demonstrates how complex projects can be designed in a fairly short period of time is the Captain device built by students at University College London. The Captain device is a cloud-connected architecture for IoT platform management comprised of a middle aggregation layer via an ultra lightweight PC (x86) device running the Lab of Things and a series of Azure connected software tools. The Captain device (shown in Figure 5) is a small computer in a discrete form factor that acts as a controller and a data aggregator for cloud services. Individual pathways to devices can be made; as well, multipath collaborations with other captains can be synchronized. As LoT manages the sensor network architecture, it is possible to run Azure applications to provide management and administration of IoT zones.

The key benefit of the introduction of the Captain Device with the Lab of Things is the flexibility of the platform and enabling use case views to management of devices in its vicinity. For example in a University building with a number of IoT sensors (ambient room temperature, light, door triggers) a number of applications could be developed from the same device to be used for different scenarios and use cases by various classifications of end users. (1) *Security guard app* – to check if the building is vacant after 7 p.m.; (2) *Student app* – to find out if a lecturer is in their room; (3) *Lecturing Staff app* – to see if the common areas are quiet for reading or check peak times to avoid; and (4) *Building Management app* – to see the energy analytics of occupancy of the building; and (5) *Subscriber company app* – to determine when the university is closed, and rooms and lecture theatres can be rented to businesses. With the Captain Device and LoT, using same-hardware different applications of IoT use cases could be easily designed.

DISCUSSION AND CONCLUSIONS

This article demonstrates the efficacy of Lab of Things as a teaching and instructional tool. The feedback that we received from the students of the smart home automation course was overwhelmingly positive. They thoroughly enjoyed the hands-on experience of working with sensors and cameras and felt they gained an in-depth understanding of how to integrate these disparate pieces of hardware into a single LoT application. However, we encountered two key challenges. First, a big deterrent to learning LoT was the programming language, C#. At UMBC, and in most universities, the students are taught Java and C/C++. While C# constructs are similar to Java and C++, getting used to developing applications in C# was initially challenging for the students. Secondly, when the course was taught in the Spring of 2014, few drivers and applications were supported. The repository of drivers and applications, however, has grown exponentially since then. One key development is support from Arduino [8]. Students, especially computer engineers, are excited about Arduino as it increases the flexibility of developing novel hardware and sensor systems in the future. ■

LAB OF THINGS WAS BUILT ON SEVERAL DESIGN PRINCIPLES THAT MAKE IT APPLICABLE TO TEACHING AND DEVELOPING CLASS PROJECTS.

REFERENCES

[1] http://www.strategyanalytics.com/default.aspx?mod=reportabstractviewer&a0=10170

[2] Villar, N., Scott, J., Hodges, S., Hammil, K., and Miller, C., .NET Gadgeteer: A Platform for Custom Devices, in Proceedings of Pervasive 2012, Lecture Notes in Computer Science, June 2012

[3] Dixon, C., Mahajan, R., Agarwal, S., Brush, A., Lee, B., Saroiu, S., Bahl, V., An Operating System for the Home, *Proc. NSDI 2012.*

[4] http://rethinkdeafhome.com/

[5] http://www.xbox.com/en-US/xbox-one/accessories/kinect-for-xbox-one

[6] http://zwaveguide.com/Z-wave-cameras/

[7] http://neurosky.com

[8] http://www.arduino.cc