Internet-of-Things(IoT) Development Stack Review

By John Koh, ECE '17

Introduction

The concept and the emergence of Internet of Things (IoT) has been a popular topic as of late. Owing in part to a rapid explosion in the availability of small and inexpensive System On Chip (SoC) and computing hardware, there is a multitude of IoT prototyping kits and development boards available to the general hobbyist. This report presents a general overview of the current IoT development scene from a hardware platform perspective. A brief overview of key features crucial for a general IoT development platform as well as ease of use is also provided. The Azure Team has considered a list of possible hardware development platforms for their design of a smart horticulture device, and this report summarizes their findings.

Problem Description

Overview

The Smart Pebble should operate autonomously to poll the sensors attached, and to periodically push the sensor readings wirelessly to a MongoDB database residing on the cloud. It should be in the form of a discrete device, being powered and operated without human intervention for an extended period. On the cloud, a server side application will then crunch the stored values to provide a customized feedback tailored to each user via a web application.

Networking Review

The ZigBee wireless protocol via the use of ZigBee routers attached to an Arduino Mega were initially selected as a candidate for the networking stack, as it offered a secure and reliable way for the devices to communicate with each other, and implementations were available out of the box. ZigBee technology is a low data rate, low power consumption, low cost, wireless networking protocol targeted towards automation and remote control applications. (C. E Sinem, 2017) ZigBee also allows a mesh network of different ZigBee devices to be formed, where the addition of more ZigBee devices would strengthen the network connection. However, a gateway is required for the device to connect to the internet, and in Team Azure's application, most prospective users would only need one Smart Pebble per household. The need for a gateway would add on additional cost for potential users, and the additional middleware would pose to be another point of failure for the developers.

Z-Wave is another possible wireless protocol (Musewerx, 2016) with similar properties as the ZigBee wireless protocol. It has the additional advantage of being an established protocol designed specifically for home automation to control applications in residential and light commercial environments. It can be added to almost any electronic device in the house, even those that is not ordinarily known to be 'intelligent'. (Musewerx, 2016). However, the need for an additional gateway nullifies the advantages brought.

A Wi-Fi based Wireless Sensor Network (WSN) has the benefits of having a high bandwidth and transmission rate, being highly cost-effective (Li, L., Xiaoguang, H., Ke, C., & Ketai, H., 2011), and highly scalable on existing infrastructure. In addition, the current Wi-Fi based IoT development systems had a more robust performance compared to the development systems for other protocols. Thus, the Wi-Fi wireless protocol were selected as the prime candidate for the Smart Pebble networking stack. As there would only be one device to setup since Wi-Fi routers are ubiquitous in a modern household, a Wi-Fi capable device would have the least possible failure points plus setup time.

Physical Review

The final product should be small enough to fit into conventional flowerpots as well as live up its moniker as a Smart 'Pebble'. At 36.58 x 20.32 x 6.86mm and weighing 5 grams, the Photon was the undisputed winner in this section as there was not much need for the extra input and output capabilities of the Arduino, nor there was any need for the increased computing power of the Raspberry Pi. In addition to that, the device's external form factor needs to be water resistant for the occasional splashes from watering, as well as being able to withstand a wide range of temperatures. Although the Photon chip by itself is not weather resistant, packaging the final product in a 3D printed case and filling any external crevices with silicone or epoxy will reduce the possibility of the device being damaged by the humidity.

Connectivity and Power Review

As per Team Azure's prototype specifications, the final product should be able to host multiple sensors, polled on an asynchronous or synchronous manner. The ideal prototype was modeled with 6 separate sensors connected to the host. If the device can process signals from multiple sensors and publish them over an encrypted channel, it is capable enough for Team Azure's work requirements. In this regard, all the tested hardware development systems had no problems satisfying the requirements. With 8 digital pins and 6 analog pins, even the Photon has enough connections to satisfy the connectivity requirements.

As the final device is to be powered by an external solar panel over extended periods of time, having low power consumption is a crucial requirement. The Raspberry Pi 2, having a more powerful processor, drew substantially more power than other choices.

Development Hardware	Average Power Consumption
Raspberry Pi 2	1.5A
Arduino Mega with Wi-Fi Shield 101	150mA
Arduino Mega with ZigBee Router	130mA
Photon Wi-Fi Development Kit	80mA

Table 1. Comparative summary of the average power consumption in different development hardware

Additional Features

To future proof the Pebble, updates should be ideally provided to consumers via the over-the-air (OTA) protocol. No physical connection to a host computer should be needed for users to get updates, thus reducing the amount of time spent by both the owner and the user of the Pebble on maintenance. There is no existing solution for Arduino Megas to be updated without connecting to a physical host, but both the Photon and the ZigBee routers has existing software infrastructure to enable over-the-air software and firmware updates.

Conclusion

The selection of a right IoT development platform for the Smart Pebble has been an important issue for the Azure Team as the hardware should be purchased and shipped over before assembly and testing could start.

The Azure Team started off by using an Arduino Mega that was connected to a host computer via a ZigBee router. The process was cumbersome as the data transmission was error prone, not to mention that the ZigBee router dropped the connection periodically. The ZigBee router were then switched out for the Wi-Fi Shield 101; however, the setup was unnecessarily bulky with the size and heft of the large shield. The data transmission to the cloud server as well as the polling of the sensors worked as expected. A Raspberry Pi prototype were also tested; however, the high-power draw of the device disqualified it from being a viable alternative. The Photon Development Kit were ultimately chosen as the best choice out of the given options above.

References

1. C. E Sinem. (2007, May). "ZigBee/IEEE 802.15.4 Summary". Retrieved November 30, 2016, from http://pages.cs.wisc.edu/~suman/courses/838/papers/zigbee.pdf.

2. Musewerx (2016). Z-Wave Wireless Control, Technology, System and Applications. Retrieved November 30, 2016, from http://www.musewerx.com/Whitepapers/Z-Wave White Paper.pdf

3. Pew Research Center. (2015) Americans' Internet Access: 2000-2015. Retrieved November 30, 2016, from http://www.pewinternet.org/2015/06/26/americans-internet-access-2000-2015/

4. Li, L., Xiaoguang, H., Ke, C., & Ketai, H. (2011). The applications of Wi-Fi-based Wireless Sensor Network in Internet of Things and Smart Grid. 2011 6th IEEE Conference on Industrial Electronics and Applications.

5. Eclipse Foundation (2016, September). The Three Software Stacks Required for IoT Architectures. Retrieved November 30, 2016, from https://iot.eclipse.org/resources/white-papers/Eclipse IoT White Paper - The Three Software Stacks Required for IoT Architectures.pdf

6. Sangeeth K., Pradeep P., Rekha P., Divya P., Aryadevi R.D., Sudheer M. (2016) Over the Air Programming Method for Learning Wireless Sensor Networks. In: Kim K., Joukov N. (eds) Information Science and Applications (ICISA) 2016. Lecture Notes in Electrical Engineering, vol. 376. Springer, Singapore