Team Azure: Smart Pebble

Remote Monitoring of the Soil Quality with the Internet of Things

By Andac Demir, ECE '17

Many industries are transformed by the IoT technologies and agriculture is also one of them. To monitor remotely their crop growth and the quality of soil in real time, quickly analyze these data combined with third party information such as weather and get better crops with less use of irrigation and fertilizer, farmers and gardeners demand to be more proactive, efficient and sustainable. The goal of Azure Team's senior design capstone project is to build a small and affordable sensing suite that offers farmers and gardeners a quick, convenient and effective way of monitoring the pH, humidity and temperature values of their soil.

Introduction

IoT devices are expected to have a significant impact on farming and gardening while meeting the increasing consumption needs of a global population that is estimated to increase by 70% by 2050 (Bughin, J., Chui, M., & Manyika, J., 2013). Companies are working to build and sell new smart agriculture IoT devices and farmers capitalize on the opportunities IoT devices bring to increase their efficiency and create/sustain a leadership role in the market. Although the IoT technologies are expanding, there is currently no off-the-shelf product that enables farmers to monitor remotely multiple properties of the soil in real time.

The objective of Azure Team is to gather data from pH, humidity and temperature sensors at regular intervals with a microcontroller and forward that data via Wi-Fi connection to the MongoDB database residing on the cloud. 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.

System Overview

Hardware Section

Hardware section can be divided into three sub-sections: microcontroller, power unit and sensors.

Our device's microcontroller (Photon) has a built-in Wi-Fi shield with ample developer support which makes it a great choice for IoT projects. For the power unit, our device uses a Li-ion battery and a solar panel. Li-ion battery outputs a nominal 3.7V at 2000mAh and includes a built-in protection against over voltage, over current and minimum voltage, which is a great feature since it protects the battery as we store the generated solar energy in it.

On the other hand, the solar panel is capable of generating 2.2W in the open sun with a peak power output around 6V at 378mA. The panel is also coated with waterproof urethane that makes the outdoor use of our device sustainable.

Our sensor suite consists of SHT10 soil temperature and moisture sensor and SEN0161 pH probe. Soil temperature and moisture sensor is cased in a sintered metal mesh that prevents water from seeping into the body of the sensor and damage it, but allows the air pass through. This sensor can even be submersed in water, which we do not actually need, but such a feature is a good sign for its durability.

SEN0161 is a cheap pH electrode and can be easily connected to our microcontroller. The disadvantages of this electrode is that it has a life time of six months and it was actually designed to measure the pH value of liquids. Had we better resources, we could have used a pH electrode with a lifetime of two years. On the other hand, when the soil is too dry, the sensor does not provide an accurate output. For this reason, after several measurements and testing, we determined a threshold level for the humidity. Below the threshold, the users is notified that the soil is too dry for an accurate pH measurement and above, there is a linear correlation between the voltage output from the pH probe and pH values:

Voltage(mV)	pH	Voltage(mV)	pH
414.12	0.0	-414.12	14.0
354.96	1.0	-354.96	13.0
295.80	2.0	-295.80	12.0
236.64	3.0	-236.64	11.0
177.48	4.0	-177.48	10.0
118.32	5.0	-118.32	9.0
59.16	6.0	-59.16	8.0
0.00	7.0	0.00	7.0

Table 1: Voltage Outputs from the pH Probe and pH Values

Software Section

After testing the sensors for any malfunction or error, we opened an event stream for our device using the Particle JS library in the index.js of the node.js application. To save the events to a log, we used a free PostgreSQL database provided by Heroku, which is an ecosystem of cloud services that offers a secure and scalable database with great developer tools. Then we added an endpoint to the node.js app to read the table from the database and display it on the web.

Communication System

Our device is able to send soil data to a server on the internet and in return to receive commands. It operates autonomously, polling the sensors attached and periodically pushing the sensor readings wirelessly to the database residing on the cloud. On the cloud, a server side application crunches the stored values to provide a customized feedback tailored to each user via a web application.

We used a Wi-Fi wireless protocol for our device's networking stack, since it has the benefits of having a high bandwidth and transmission rate and being highly cost-effective (Li, L., Xiaoguang, H., Ke, C., & Ketai, H., 2011). Besides, the current Wi-Fi based IoT development systems are more robust than the development systems with other protocols.

Our microcontroller Photon has an existing software infrastructure that enables over-the-air software and firmware updates. Updates to our device is provided via the over-the-air (OTA) protocol. No physical connection to a host computer is needed for users to get updates, which significantly reduces the amount of time spent by the customer on maintenance.

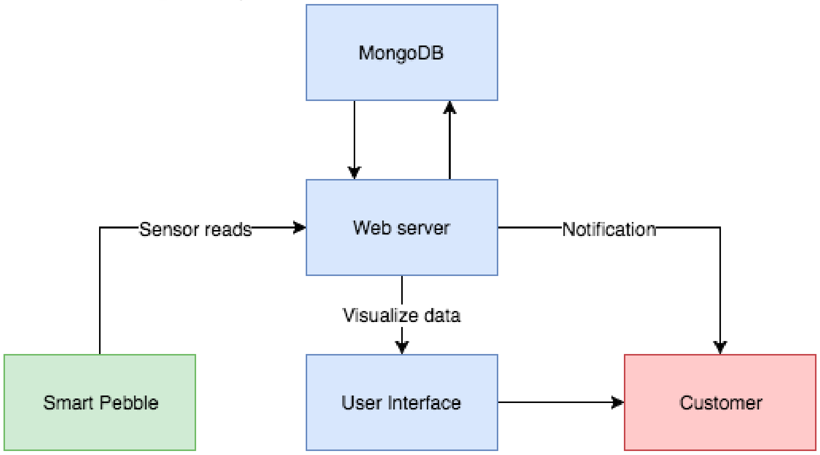

Figure 1: Systems Diagram of Smart Pebble

Acceptance Tests

Hardware Acceptance Test

The microcontroller was the easiest to test. It simply required us to write basic code for polling pins and transmitting the readings over Wi-Fi.

The sensors required more effort. Each sensor was tested using soils with different humidity, temperature and pH properties. We determined if the sensors can pick up any deliberate changes to the physical conditions of the soil and how long it takes to do so. Each sensor is then calibrated with respect to the test results. The soil we used in tests was chernozem, a black soil, containing high percentages of humus, nutrients, ammonia and acids and the tests were carried out in the city of Boston and inside a building.

We tested the power module in the bad winter weathers of Boston and we concluded that the Li-ion battery and the solar panel can sustain the power our device needs for more than 3 months in the worst case scenario; however with further product development and testing, the battery life can be enhanced easily and there is a big room for improvement. Furthermore, the customer is notified when the device runs out of power and then he can recharge the Li-ion batteries with the solar panel.

Software Acceptance Test

a) Back-end Test

Back end test is the server side test of our device. Back end development is very critical because it is what makes the front end possible. Back end questions consist of where all the data will be stored and how our software team will handle the problems with the database.

One way to test the back end side was to connect multiple devices to the cloud and store and process data for long times. That way we can be sure if the server faces any significant problems and if it is constructed properly; however we do not deal with big data currently and therefore this test was neglected. Nonetheless, the data from a single device was monitored for long times and no unusual situations occurred with the flow of data: such as loss of data, long delays between data transmission and reception or server communication failure.

b) Front-end Test

Front end development is the client side, so the user can see the data readings processed by the web application and interact with the device directly. The most important acceptance test in the front end site is the quality of data visualization and user interface. The user is able to login to our web app either with his Facebook or email.

First of all, data visualization of our device meets the need for speed; however even if we find and analyze data quickly and put it in the proper context for the user, the value of data for decision making purposes will be jeopardized if the data transmitted from the device is not accurate or timely. We can pass the test in data visualization only if we can assure the data quality (Chen, C. P., & Zhang, C. Y., 2014). To avoid this problem, in addition to making several hardware tests to calibrate our sensors and get correct measurements, we also created a proactive information management system to ensure that the data is clean.

Besides, charts and plots were made easy to read and display meaningful results. We will handle the problem of processing big data that accumulate over time by clustering the data into a higher-level view where smaller groups of data can become visible (Chen, C. P., & Zhang, C. Y., 2014). By grouping the data together, or "binning," we will be able to visualize the data more effectively; however since our data is not that big, we skipped this step.

c) Integration Test (Full stack development)

A meticulous unit test system was integrated before the demonstration of our prototype, and various facets of the functionalities were tested such as: the layout time, load time and user-friendliness.

Conclusion

Our device, Smart Pebble, promises to deliver 24/7 visibility into soil allowing farmers and gardeners to track the levels of humidity, temperature and pH. With the attributes that include accurate sensor readings, rechargeable power module, waterproof casing, user-friendly data visualization and over-the-air firmware updates, it increases the efficiency for the farmers and improves the decision making process of the gardeners.

References

1. Bughin, J., Chui, M., & Manyika, J. (2013). Ten IT-enabled business trends for the decade ahead. *McKinsey Quarterly*, *13*(May).
2. Li, L., Xiaoguang, H., Ke, C., & Ketai, H. (2011, June). The applications of wifi-based wireless sensor network in internet of things and smart grid. In *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on* (pp. 789-793). IEEE.
3. Chen, C. P., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*, 314-347.