Progress Report
E-Garden System

Project Team members: Timmy Kwok
Duling Lai
Weidi Zhai
Siyan Chen
Bo Sun
Tianguang Zhang

Contact Person: Timmy Kwok
sumyuek@sfu.ca

Submitted to: Dr. Andrew Rawicz
Steve Whitmore
School of Engineering Science
Simon Fraser University

Date: Nov 28, 2015

1.0 Introduction

The E-Garden is aimed to bring the best convenience to maintaining garden. Aside from automatically watering the plant, our product also collects and displays soil humidity and temperature data of your garden. Our web APP, which includes a web server and web interface, will enable you to maintain your garden anytime from anywhere. Our development of E-Garden proof of concept model has focused on three major components; the progress of each will be introduced in the document:

- Hardware development aimed to provide a real time feedback system
- Firmware development on the microcontroller to achieve digital packages transmission through Wi-Fi and Bluetooth network to multiple sensors
- Software development aimed to develop a web server to control the E-Garden

2.0 Progress Summary

2.1 HARDWARE PROGRESS

The hardware part of this project consists of the DHT11 temperature sensor, the YL-69 soil moisture sensor, the HC-05 Bluetooth module, the servo motor, the Arduino Uno microcontroller and a power supply. The sensors, the Bluetooth module servo motor and the microcontroller have been assembled on a solder-less breadboard with wire connecting to the power supply. The DHT11 temperature sensor works well; the microcontroller can receive the digital signal from it and send this signal to the raspberry pi by the Bluetooth module. However, there is a problem of the YL-69 soil moisture sensor, the output signal is an analog signal, and the microcontroller cannot read the humidity directly. For the servo motor part, our goal is to control the valves by using a servo motor which has sufficient torque force to rotate the valves. However, there is a problem that still needs to be solved. In order to control the servo motor, Arduino Uno has to receive a different signal from Raspberry Pi to make different action. For example, if Arduino Uno receives a signal 1 from Raspberry Pi, it will send a command to servo motor. Hence, the servo motor will turn on the valves to watering plants. Conversely, if signal 0 is received, the servo motor will turn off the valves. Finally, we have solved the power supply problem. By using rechargeable battery, Arduino Uno can be easily powered and installed in any place in the garden.

2.2 SOFTWARE PROGRESS

The software team has been working on the web application which is hosted on Amazon Web Service. The website is set up and running with features including login function, displaying temperature and moisture level and manual watering function. The connection between the MySQL database and Raspberry Pi is well handled. The implementation of plant library and historical data is delayed, as the watering function takes longer to complete. Tests have been done to ensure the functionality of the website and its connection to Raspberry Pi. User meetings are planned, and will be organized once the whole system has been successfully integrated.

2.3 FIRMWARE PROGRESS

The firmware section has been working on the Raspberry pi connection. Bluetooth and Wi-Fi connection is the major two connections between the device and a webserver. We have created the network that is able to insert data from Raspberry pi to web server, and we had also setup our Bluetooth connection with different devices. Furthermore, we have now successfully input data from Raspberry pi to a web server which can let the user able to read the temperature and moisture level on their own web page. However, we still have to modify data transfer by using Bluetooth, as temperature and moisture cannot successfully transfer to raspberry pi. As we researched that the major bug that we are facing on Bluetooth, the language that we programmed in Arduino and Raspberry pi is different and the value that the present temperature and moisture level have a different name. Therefore, we have to fix the problems and look for other solutions to achieve the product.

3.0 Remediation

3.1 HARDWARE REMEDIATION

First of all, we need to write a program in the microcontroller to transfer the analog signal from the moisture sensor to digital signal, so that the microcontroller can read the real humidity and send it to the raspberry pi. Then, the sensors, Bluetooth module and servo motor have to be soldered onto the microcontroller to reduce the volume of the enclosed case. Finally, we have to combine all the hardware parts together and to see whether it works well or need to do any further adjustments. We are a little bit behind the schedule due to the coding. Therefore, we decide to test all the hardware part and Raspberry Pi by December 4th.

3.2 SOFTWARE REMEDIATION

The plant library feature will be implemented on a separate web page where users can set the plant type and watering amount. However, this requires writing a new webpage which may take a large amount of time. In order to meet the presentation schedule, this feature may not be implemented for the demo in the worst case. Depending on the schedule, the historical data page may not have a plot graph, but a simple table of historical data instead.

3.3 FIRMWARE REMEDIATION

To fix the problem that we are facing, we might have to design another planning, in order to connect the sensors and servo motor to our Raspberry pi. During the semester, our group had designed two ideas that can fix the problem. The worst case we can only connect the sensor directly to the Raspberry pi with wire; however, it might makes our costumers uncomfortable. The second case we can setup Wi-Fi connection from Arduino to Raspberry pi, in order to send out and receive information. However, users might need to keep charging the battery every day. As we are having limited time left, we are planning to focus on the Bluetooth part and Wi-Fi connection part because these are the most possible connection that we can finish.

4.0 Schedule

The initial plan was to finish hardware and software design before Dec 1st. However, due to technical problems when combining all hardware and software together, we are currently a week behind schedule. As of now, we will proceed with the following schedule outline:

Nov 26th -to- Dec 1st: Focus on reliability testing
Dec 1st -to- Dec 5th: Hardware and software feature enchantments
Dec 6th -to- Dec 7th: Modular integration of whole system
Dec 7th -to- Dec 17th: Integration testing and demo preparation

5.0 Finances

Smart Garden Inc. has received funding from the Engineering Student Society Endowment Fund (ESSEF) in the amount of $375.00. We are currently well under our initial budget of 374.83. Since it is our primary source of funding, this fund was used to purchase major hardware components. By the completion of our model, our project may go over budget if there are unforeseen costs, such as replacing broken components, etc. The current breakdown of parts used is shown below.

Item	Expected Cost	Actual Cost	Difference
Raspberry Pi 2	65.99	$123.14	-$57.15
Soil moisture sensor	$14x2	$7.5	+$20.5
Temperature sensor	$4.99x2	$7.5	+$2.5
LCD display	$23.99x2	-	+$47.98
Micro Servo Motor	$14.99x2	$14.99	+$14.99
Plant water Sprinkler	$5x4	-	+$20
Small water pump	$21.45x2	-	+$42.9
Wi-Fi USB Adapter	$9.99	$7.99	+$2
SD card	$9.99	-	+$9.99
Arduino	-	$33x2	-$66
Bluetooth	-	$25	-$25
3D-Printing Encloser	-	$60	-$60
Battery	-	$10	-$10
Other Cost	$60	$30	+$30
Shipping	$50	-	+$50
Totals	$374.83	$352.14	+$23

Table 1: Expect and Actual Cost

6.0 Conclusion

In summary, we have made considerable progress on this project. Although the overall development progress is slightly behind our original schedule, we are optimistic that we will have a functional project to show on our demo day. For the finances, our overall budget estimates were very close to our prediction.