FarmSmart – An Internet of Things Application for Agriculture

Justin Kim, Tram Lam, Efai De Leon, Junning Li
[#]*Electrical Engineering and Computer Science, University of California - Irvine*
Under supervision of Stuart Kleinfelder, Professor

***Abstract*—The agriculture industry is an area where Internet-of-Things technologies can be applied in order to improve the industry. *FarmSmart* is a project with the goal of providing IoT technologies to farmers with the goal of providing data of their farm and delivering that data and analytics to an accessible interface via Cloud computing and mesh network sensors. Currently, in order to achieve this goal, the low-power ESP-8266 and Adafruit sensors are used to record data of soil and send the telemetry to an AWS cloud server, where the server will analyze the data and provide a web interface for users. The data is able to be sent successfully from a few microcontrollers and can be seen on the web interface for viewing, though the mesh network and other web features are still in development. This paper will provide the specific details of each component in *FarmSmart*. Furthermore, this paper will discuss ABET comments with regards to technical standards, constraints, and security issues.**

***Index Terms*—IoT, internet of things, agriculture, microprocessor, sensors, mesh network**

I. Introduction

THE concept of Internet-of-Things (IoT), an idea of connecting any device to the internet, has been a topic of conversation that has rapidly grown in the past few years thanks to technological advances in wireless communications and technology. With microcontroller and sensor technology becoming smaller and more power efficient, we see continuous development and deployment of IoT technologies in many industries. The agricultural industry is one such industry where IoT technology is continuously being adopted and deployed. For farmers to maximize yields and produce more food, farmers need tools to provide agricultural information and pertinent knowledge in order to make more informed decisions with regards to their farms.

In order to be one solution that provides such tools, *FarmSmart* was developed to provide data and analytics to farmers with the goal of providing full data of the entire farm. *FarmSmart* uses low-power microcontrollers and sensors to measure critical data such as soil moisture and air temperature to analyze and inform farmers of what the condition of the farm is along with suggestions on what actions to take. The main feature of the microcontrollers is the ability to connect all the sensors via a mesh network, expanding the range of each sensor. While IoT has been expanding throughout the agriculture industry, not many solutions exist for incorporating mesh networks to agriculture and IoT solutions, with the main problem being power efficiency and battery life. Algorithms to conserve power and transmit data effectively are devised to overcome the issues of mesh network IoT devices, allowing *FarmSmart* to be one of the few that is developing mesh networks in agriculture.

The progress within *FarmSmart* has been steady and promising, with rapid development towards its server infrastructure and sensor development. End-to-end communication between the nodes and the server has been established and functionally working, with the mesh network between the nodes in steady development and will be discussed further.

II. Project Details

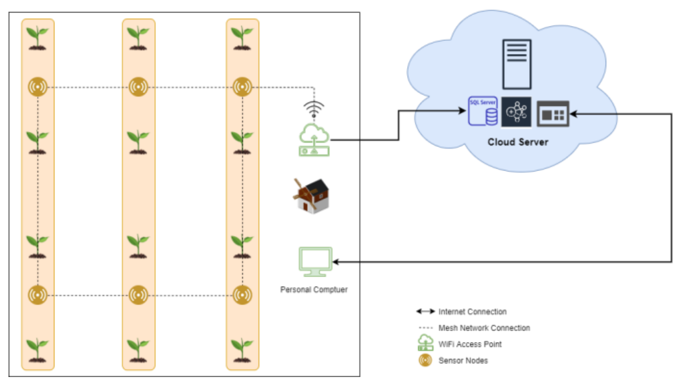

Fig. 1. Concept overview of *FarmSmart*. Sensors are placed throughout the farm to send data to a cloud server for analysis and viewing.

The main goal of *FarmSmart* is to provide telemetry regarding the farm's vitals to the farmers in order to assist the farmer in making informed decisions to manage their farms. To do this, several microcontrollers would be placed throughout the farm in key areas. These microcontrollers would be sending information, such as soil moisture, air temperature, and humidity, to a cloud server. Databased within the cloud server

This work was supported in part by the University of California – Irvine School of Engineering with financial support under EECS 159A – Senior Design.

Justin Kim, Tram Lam, Efai De Leon, and Junning Li are all affiliated with the Electrical Engineering and Computer Science department in University of California, Irvine.

would store the information received by the microcontrollers, where the data would be visualized (and analyzed in the future) for farmers to make informed decisions. This would then be viewable to the farmer by sending this data in a designed user interface (UI).

The development of *FarmSmart* is divided into several components due to the complexity of the project. Hardware must be chosen with consideration to wireless communication and power efficiency. Furthermore, a software solution must also be devised to provide a robust server and an intuitive UI for farmers to use.

A. *Software Component*

The website integrates HTML, CSS, and python programming language through "Django." Django operates a website's CSS and HTML files as templates allowing python programs to manage the site's security, scalability, and other features. The website functions as the main application to display information for each account. The homepage provides summarized data that gives a quick overview of the current state of the farm. The navigation bar links the user to 5 different web pages where they can acquire further information or create an account.

CSS enables the website to scale accordingly to the screen size. It can run on several devices with smaller screens such as smartphones and tablets. Elements will move and adjust to the screen ratio improving accessibility and readability. Furthermore, creating an account and verifying credentials is powered by forms. A Django python program, "forms.py," creates and handles each one by storing the information in a class attribute. It also restricts the character length and sets the input type for each box in the form. "views.py" handles template and form "get" and "post" operations. It defines HTML files as classes with a "get" method that returns the requested site and forms element, if available, and a "post" method that redirects to another site and submits available form elements.

In order to program the firmware for our microcontrollers, the Arduino framework for our microcontroller was chosen (details of the microcontroller and hardware will be discussed further in the following section). The Arduino Core was chosen as it has been updated regularly by the manufacturer in order to provide full functionality of the chip. The Arduino Core and flasher allows us to program the microcontrollers in C/C++, giving tighter control of memory constraints and lowering performance overhead compared to the default Lua language that can be used with the microcontroller. Free license libraries from the Arduino community allow us to develop the firmware even further in order to push a prototype within a fast timetable. The libraries allow us to have easier communication with the attached sensors.

B. *Hardware Component*

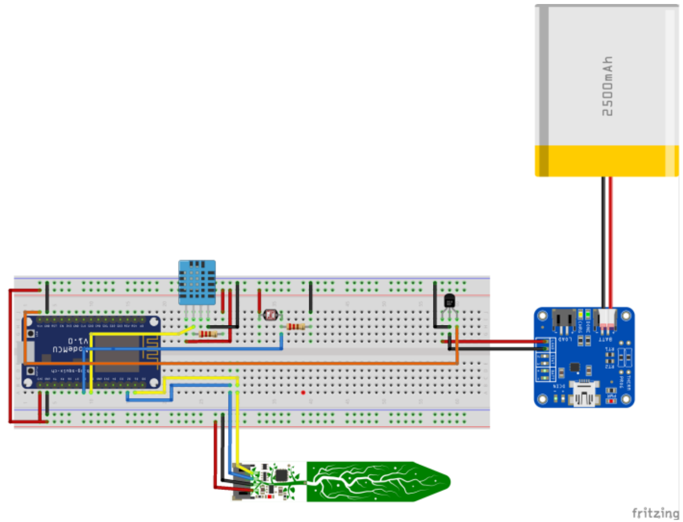

Fig. 2. Breadboard circuit diagram of the hardware to highlight the use of the parts described in this section.

FarmSmart is an IoT service that provides data analytics of farm plots in a convenient fashion for farmers through a website. In order to display unique data from individual farms, the hardware component of this project provides several different sensors which gathers information of different aspects of the farm and updates the farmer on the website throughout the day at a timely interval.

The major component of the hardware design is the microcontroller ESP8266, which is a system-on-chip that integrates a 32-bit Tensilica microcontroller, standard digital peripheral interfaces, antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management in one [1]. As a team, we specifically chose the ESP8266 because it provides capabilities for 2.4 GHz Wi-Fi, general-purpose input and output ports, inter-integrated circuit, analog-to-digital conversion, and a Serial Peripheral Interface. These features are powerful in providing better services to farmers by collectively collecting data and transferring data onto the server to display a message that is easily understandable by farmers. Currently, the team is looking to develop and expand the size of the mesh network around the farm by adding 3 more nodes to achieve a mesh network of 4 different nodes with similar sensors around the farm to return and update data on a farm collectively.

The microcontroller ESP8266 is currently powered by a 3.7 Volt 2000mAh lithium Ion battery, which is a thin, light and powerful battery that provides power not only to the microcontroller but also to the sensors of that specific node. As a group, we chose lithium battery because it is efficient and has a long lifespan. However, our future plan is to integrate and efficiently power all the nodes with solar panels and use solar panels to charge the lithium battery to obtain data from the sensors. Currently, there are three sensors in our hardware component: a photoresistor, a DHT 11 sensor, and a soil sensor. The photoresistor provides data about the light intensity of the farm, which is useful for farmers that are growing light sensitive crops either indoor or outdoor. The DHT 11 sensor returns value about the temperature and humidity values of the farm which can be useful to farmers in deciding the best course of action for their crops under those specific conditions. The team chose the Adafruit STEMMA soil sensor which is a I2C capacitive moisture sensor because unlike the other soil sensors that are resistive, this capacitive design provides only one probe that does not have any exposed metal and does not introduce any DC currents into the plants. This capacitive soil sensor provides data about the moisture level of your soil and the ambient temperature of the soil.

C. Methods

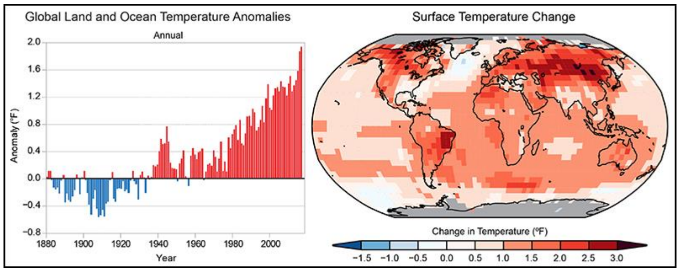

Fig. 3. Global Annual Average Surface Temperature from 1880 to 2017. Changes in climates are factors to consider and is a factor towards the development of IoT technologies in agriculture.

As depicted in Figure 1, global average surface air temperature has drastically increased by 1.8 °F since 1901. Prior to 1920, however, there has been a subtle variation in average surface air temperature and reports have indicated that within the next couple of decades. This number can be expected to increase by 2.5 °F. Further reports have indicated that such changes in the environment can bring about more natural disasters, including floods, heat waves, etc [2]. More individuals have become concerned about the effects of climate change and whether there will be enough food to feed the increasing population. With the population expecting to grow from 7.2 billion people to nearly 10 billion people by 2050 [4], researchers are looking into ways to minimize climate change effects on agriculture. Our product assists with such goals by helping farmers maximize their yields by improving the efficiency of their farms.

Many developers have attempted to introduce IoT into agriculture to help combat problems with climate change. However, *FarmSmart* is the only product that has a detailed plan of its features and how it plans to use said features to alleviate agricultural problems. *FarmSmart* detects anomalies in soil temperature and humidity and immediately alerts the farmer. The farmer can then proceed to the specific area of his farm to determine what exactly went wrong before something detrimental can happen to his crops. Since farmers own acres of farmland, it can be difficult to pinpoint the exact location that is not fully function. Therefore, *FarmSmart* aims to alleviate such worries by showing farmers where and when their farm is having issues. This increases the yield of their produce and helps farmers grow more crops with less farmland.

While data is an important part of maintaining a farm, it is important to know how to efficiently use that data. *FarmSmart* does the thinking for the farmer. It uses the data it obtains and translate it into text and visuals that farmers can understand through its user interface. In IoT based farming, data is observed, analyzed, and decisions are suggested to the farmer on how to efficiently maintain their farm. *FarmSmart* focuses on precision farming and making sure the farm is in a controlled environment. Doing so will prompt farmers to quickly react to any emerging issues they may see within their farms.

D. Results

The progress of *FarmSmart* has been very promising as our team currently have our nodes properly obtaining data from its sensors and sending it to our server. Each node will include a soil temperature and humidity sensor and an air temperature sensor. For our prototype, data is currently being sent to the server every hour in a text string. This is string is then parsed into different section based on what type of information it referencing to. In the upcoming weeks, we plan to work on establishing a mesh network between each node. Ideally, we want to have only one node send the relevant data to the server every hour.

While the backend manages the obtained data, the frontend side works on outputting that data onto a website. A basic website has been designed on HTML and is currently online for those who wish to view it. The website can be accessed on both smaller and larger devices, making it easily accessible to farmers who are in home and out on the field. It currently contains a login feature to differentiate between different users and nodes. Additionally, there is an interactive graph that will change according to what the user wants to see and what data is being sent. The website is also updated every time new information is sent to the server.

In the upcoming weeks, we also plan to improve upon our website by adding additional features such as notification alerts and other visuals that may help farmers control their farm better. The team is also looking into integrating weather and other useful APIs to inform the farmers about any sudden change in climates in the upcoming days. Our development plan has been very steady and we are working on bringing more innovative ideas to our product to help assist farmers maximize the efficiency in their farms with low power and cost.

III. Conclusion

In order to produce crops efficiently, farmers need access to information regarding their farms and crops. With advancements in IoT technology, projects like *FarmSmart* are able to provide these farmers with sensors that can provide data and analytics at a constant rate. In order to achieve this, low-power microcontrollers like the Espressif ESP-8266 can deployed in a mesh network to measure farm vitals and upload the telemetry to a cloud server. From this server, farmers are able to access the vitals via a website with an intuitive UI.

FarmSmart's current state allows the data from the microcontrollers to be sent successfully to the cloud servers, where that data can be seen in a graph on the website. Other features like a login system are integrated to the server for a more feature-complete software service. As previously mentioned, we plan to integrate more features that will provide farmers with useful information and improve the data analytics, with features such as weather station integration and notifications in future development. Also in the future are further plans to develop the hardware even further, potentially including more sensors and to improve the mesh network to be able to triangulate the positions of the microcontrollers within the farm.

APPENDIX

A. Appendix 1

Complying to standards is beneficial for many reasons. It allows team members to communicate with each other transparently and refer to industry-backed resources and references. It also allows for easier communication and product design with other teams as there is a common language to reference. This is useful not only when developing the project, but also when interfacing several components such as from a microcontroller to a server.

SmartFarm is a project that complies and will comply with many standards, particularly through wireless communication. In regards to the WiFi module, the ESP-8266 microcontroller follows the TCP/IP protocol and IEEE 802.11 WiFi communication standard, specifically the 802.11 b/g/n WLAN MAC protocol [JK1]. This standard was chosen by the manufacturer of the microcontroller as it is the industry standard for WiFi technology, with many access points and routers following the same standard, and is vital for easier and proper communication with any device that these microcontrollers communicate with. The 802.11 standard also allows the implementation of mesh networks via WiFi, a feature that was enabled in this specific microcontroller and a reason why this microcontroller was chosen.

The communication between the microcontroller and the server also follows the HTTP protocol with proper message formatting in order to ensure proper communication with the microcontroller and the World Wide Web [jk2]. By following this protocol, it is much easier to diagnose any connection and reliability issues. It also gives us the flexibility of choosing how to format the data being sent to the server with the ability to attach any message necessary with HTTP communication.

In order to improve the functionality of the device and improve security with the HTTP communication, an encryption standard like AES-256 encryption may be implemented to tackle the issue of security. Implementing a form of encryption is considered as the issue of security with data must be considered, and while the encryption is handled with regards to the server and the website, the same standards should be implemented in the communication between the microcontrollers and the server. The reason why AES-256 bit encryption is considered is that AES is a federal standard and included in the ISO/IEC 18033-3 standard. [jk3]

Currently, our devices are compliant with many of the standards listed above (except for AES-256 bit encryption) as the project would simply not be able to function if it was unable to comply to these standards. Communication would not work properly as bad request formats would occur from the cloud server if the microcontroller did not format HTTP requests properly. Non-compliance with the 802.11 standard would not allow communication between the microcontroller and other network-based devices.

B. Appendix 2

As a team, *FarmSmart* came across lots of obstacles in different aspects as we try our best to develop a convenient and valuable product for our future users. One constraint we faced in developing this project was searching and settling for a reachable goal that we can accomplish within the two quarters. As a team of four, we had lots of different ideas, however, not all of them are accessible. After talking and listing all the pros and cons of each project idea and calculating the possibility of developing and implementing the different ideas, we decided on focusing on *FarmSmart* as our project. After settling for a project, we ran into another issue: scheduling conflicts. It is quite difficult to work together as four due to different class schedules and plans. However, we all have a common understanding that we must work together in order to make our project idea a reality. The team then decides to have weekly meetings where we update each other of our own progress and what our plan for the next week are. During these weekly meetings, we also discuss our personal challenges on the project and try to help each other to solve them simultaneously.

The biggest challenge the hardware component of *FarmSmart* faced was cost and funding. Fortunately, with the help of the EECS department, we were able to obtain most of the hardware parts we needed. With the given funding, we were able to obtain four soil sensors, four lithium batteries, and a solar panel. Our goal is to have a mesh of nodes around the farm, for testing purposes, four nodes is enough to create a protocol, so we are grateful in obtaining the parts with the given funding. However, we also had to cut down a lot of other parts we wanted because there is a limited amount of funding. If funding was provided, we would have loved to have more sensors to help to get more useful data for farmers and we would have loved to have solar panels implemented at all the nodes we have around the farm.

The biggest challenge the software side of *FarmSmart* faced was choosing and deciding the form of user interface we would like to implement for our project. Marketability was one of our major concerns in choosing the form of user interface for our project. Since our targeted users are farmers, we wanted to develop a user interface that would be the most convenient and easy to use and read for farmers. The options were between a phone app or a website. The team collectively agreed to develop a website in this case with user log in systems for better interactive graphs and easier access for farmers because a website is accessible on mobile and on a computer. Also, a website would be accessible from any brand of phone while phone app might be limited to the type of phone the farmer has (iOS applications can only be accessed from iPhones and Android apps can only be accessed within Android products).

C. Appendix 3

Security vulnerabilities exist in any online application. Our software exhibits potential security risks in data transferring and credentials authentication. Our website operates on Amazon Web Services (AWS) that also present security issues to our software. AWS handles most of the security issues regarding the server, but we have taken additional methods to protect our account. For instance, it has a group of users to handle identity and access management to detect potential security threats. To mitigate this fault, we also enable a multi-factor authentication method and tools useful to troubleshoot security issues. Furthermore, Django, a python library that integrates HTML, CSS, and python programs to develop online applications, handles actions such as storing data into our local SQL database and validating credentials forms. For example, Django has a built in SQL injection protection, which is a malicious user attack that executes unsafe SQL code on a database. According to Django, “Django’s query sets are protected from SQL injection since their queries are constructed using query parameterization. A query’s SQL code is defined separately from the query’s parameters. Since parameters may be user-provided and therefore unsafe, they are escaped by the underlying database driver.” [ED1] Consequently, this security feature prevents data leakage and data deletion from our database.

On the other hand, IoT hardware promises to connect anything to the cloud. However, security issues are imminent in such an undertaking. Data coming from the sensors are susceptible to interception and leakage by a malicious third party. To mitigate the threat, we are planning to implement secure encryption on our sensor data communication. Many security risks exist in IoT projects as sensors run for an extended time with limited security management. However, to reduce this issue, we notify the user when erroneous data is more prominent so that they can update the modules or replace the sensors.

ACKNOWLEDGMENT

We would like to thank Professor Stuart Kleinfelder for the guidance and funding for the project. Without the guidance of Professor Kleinfelder, the project would not have been able to achieve the development progress it has been able to get to along with the funding necessary to get the first prototype working properly.

We would also like to thank Beverly Abadines Quon, PhD student, for providing additional direction on where to deliver this project along with feedback on how to develop and improve the project. We would also like to thank her and the rest of the EECS 159A team for providing laboratory equipment in order to diagnose issues.

REFERENCES

[1] "ESP8266 Overview | Espressif Systems", Espressif.com, 2019. [Online]. Available: https://www.espressif.com/en/products/hardware/esp8266ex/overview.

[2] D. Wuebbles, "How Will Climate Change Affect the United States in Decades to Come?", Eos, 2019. [Online]. Available: https://eos.org/features/how-will-climate-change-affect-the-united-states-in-decades-to-come.

[3] B. Plumer, "Can We Grow More Food on Less Land? We’ll Have To, a New Study Finds", The New York Times, 2019. [Online]. Available: https://www.nytimes.com/2018/12/05/climate/agriculture-food-global-warming.html.

[4] ESP8266EX Datasheet. Shanghai: Espressif Inc., 2019.

[5] V. Beal, "HTTP - HyperText Transfer Protocol", Webopedia, 2019. [Online]. Available: https://www.webopedia.com/TERM/H/HTTP.html.

[6] "Advanced Encryption Standard", Wikipedia, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard#References. [Accessed: 09- Dec- 2019].

[7] "Security in Django", Django, 2019. [Online]. Available: https://docs.djangoproject.com/en/3.0/topics/security/. [Accessed: 09-Dec- 2019].