A Smart Sensor Network for an Automated Urban Greenhouse

Kala Meah
Dept. of Engr. and Computer Science
York College of Pennsylvania
York, PA, USA
kmeah@ycp.edu

Jason Forsyth
College of Integrated Science and Engr.
James Madison University
Harrisonburg, VA, USA
forsy2jb@jmu.edu

James Moscola
Dept. of Engr. and Computer Science
York College of Pennsylvania
York, PA, USA
jmoscola@ycp.edu

***Abstract*—An effective and efficient sensor network is an essential component of an automated urban greenhouse (AUG). This paper describes a design process that prototyped a smart sensor network as part of a capstone design project. This smart sensor network communicates among sensing, power and automation, and visualization and user interface aspects of the AUG to provide automatic monitoring of lighting, heating, watering, and ventilation. This automated urban greenhouse, along with the sensor network, is being installed at a local elementary school in downtown York, Pennsylvania. It will serve as an educational tool and create awareness of the importance of fresh foods.**

***Index Terms*—greenhouse, sensor network, automation, K-12 education**

I. Introduction

The United States Department of Agriculture classifies York City, Pennsylvania, USA as a “food desert” which is defined as an urban area where it is difficult to purchase affordable or good-quality fresh food [1]. To address this challenge, capstone engineering students at York College of Pennsylvania designed an automated urban greenhouse to provide access to fresh produce and support science education at a nearby elementary school in downtown York, Pennsylvania. Central to this project was a smart sensor network to automatically monitor and maintain lighting, heating, and ventilation in the greenhouse. This sensor network had the following design objectives:

- maximize electrical energy efficiency;
- manage heating, cooling and ventilation of the greenhouse;
- use historical and current weather data to estimate future greenhouse energy needs;
- collect and present greenhouse information (temperature, humidity, ambient light, etc.) to elementary school students and teachers through an accessible mobile or online interface; and
- allow for manual overrides of all automated functionality in the greenhouse.

It was also important to consider that this greenhouse is designed for use by elementary school students. Special attention was paid to ensure that all components were both safe and child-friendly. Any materials with sharp edges, harmful chemicals, or high voltages were strictly prohibited from areas that students would occupy.

To determine the project’s specifications, the capstone design team took multiple trips to the elementary school to meet with the principal, members of the maintenance staff, and representatives from local organizations who volunteer to maintain the school’s courtyard. After interviewing everyone, the capstone design team was able to further understand what the end users were expecting from the project and how to create specific requirements for each goal. A 3D model of the final greenhouse design is shown in Fig. 1. The sensor network was implemented by three teams of students with each team focusing on one of the three subsystems: sensing, automation, and data visualization. This paper describes the hardware and software implementation of the sensor network, its subsystems, as well as communication protocols for the automated urban greenhouse.

II. Related Work

Smart sensor networks for greenhouse applications have been an active research field for the past couple decades. In recent work, Kannan and Thilagavathi present a Zigbee-based wireless sensor network for monitoring environmental conditions of farms [2]. Their work utilizes microcontrollers to collect sensor data and wireless cameras for a live video feed, all accessible over the internet. In [3], Satpute and

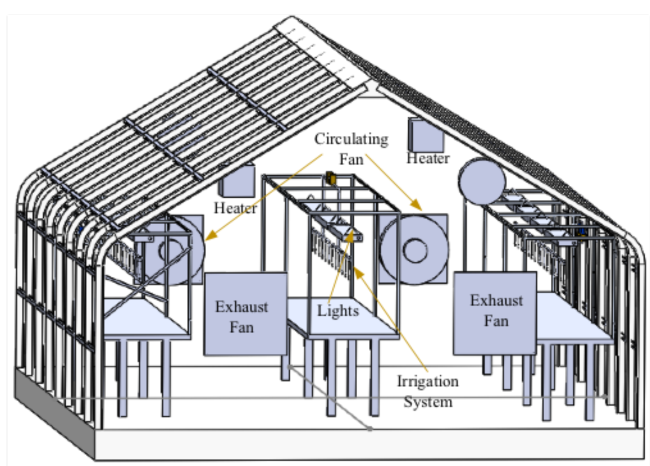

Fig. 1. 3D model of the greenhouse

978-1-5386-8014-8/19/$31.00 ©2019 IEEE

Theng implement a wireless sensor network for monitoring temperature and humidity in an industrial environment. Wireless sensor nodes transmit sensor data to a central node that displays sensor data on an LCD. Ferentinos, et al. deployed a prototype wireless sensor network in a commercial greenhouse to analyze the reliability and accuracy of sensor nodes [4]. They noted that solar radiation dramatically affects the accuracy of sensor nodes. However, shading sensor nodes with simple metallic shields provides sufficient protection from solar radiation to produce accurate sensor readings.

III. Design of Greenhouse Subsystems

This section describes each subsystem of the sensor network along with integration and implementation.

A. Sensing Subsystem

The focus of the sensing team was to design and implement sensing devices to gather temperature, humidity, light, and moisture data to determine the current state of the greenhouse. After collecting data, the sensing subsystem sends the data to a visualization subsystem so it can be logged and viewed via the greenhouse webpage. The visualization subsystem sends sensor data, as well as other control signals, to the power and automation subsystem. The power and automation subsystem uses the data and control signals to control greenhouse equipment such as heaters, lights, water bulbs, and shade.

Design and component selection focused on the following: cost, ease of use, durability, and accuracy of sensor readings. Given that the greenhouse was designed for use by primarily elementary school children, durable, low-power, and child-friendly sensing device were selected for the design. The accuracy of the sensors is very important for the correct operation of the automated greenhouse systems. All sensor selections went through a selection matrix and the following sensors were selected in each category: light sensor – Adafruit TSL2591 [5]; humidity sensor – Adafruit AM2302 [6]; moisture sensor – Decagon Devices 10HS [7].

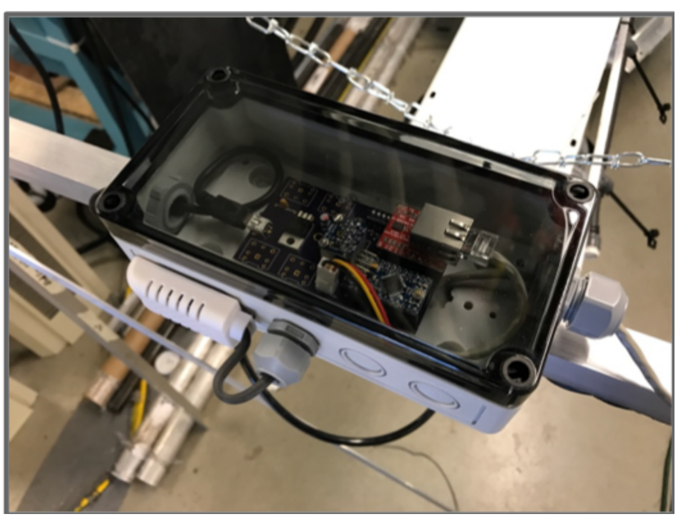

Fig. 2. Temperature, light, and humidity sensor enclosure

The automated urban greenhouse has six individually controllable moisture zones, each with two moisture sensors for a total of 12 moisture sensors. The greenhouse design includes five each of the temperature, light, and humidity sensors distributed throughout the floorplan. To house and protect all of these sensors, two different types of weather-resistant enclosures were assembled: moisture sensor enclosures and temperature, light, and humidity (TLH) enclosures. Three of the moisture sensor enclosures were assembled, each able to connect to and get readings from four moisture sensors. Five of the TLH enclosures were also assembled. Each TLH enclosure has a clear top to allow light to enter. A temperature and humidity sensor is mounted on the exterior of each TLH enclosure. A photo of the TLH enclosure is shown in Fig. 2.

Both the moisture sensor enclosures and the TLH enclosures include an Arduino Pro Mini microcontroller to read data from the various sensors. Additionally, both enclosure types include an Olimex ENC28J60-H Ethernet controller connected to the Arduino. The ENC28J60-H Ethernet controller was selected due to its low power, small footprint, and Arduino compatibility. A common Printed Circuit Board (PCB) backplane was designed for use in both the moisture sensor enclosure and the TLH enclosure. The PCB was populated with the appropriate components for the type of enclosure in which it was used. For example, no TLH components were populated on the PCBs installed into moisture sensor enclosures. The Arduino in each enclosure collects new sensor data every 20 seconds. Sensor data is assembled into a JSON formatted message and sent to the visualization subsystem in User Datagram Protocol (UDP) datagrams over an Ethernet network.

B. Visualization and User Interface Subsystem

The objective of the visualization and user interface team was to collect sensor data and to develop an easy-to-understand user interface on which elementary school teachers, students and administrators could view that data. This team was also tasked with creating a control interface that would allow these users to control the greenhouse while inside the greenhouse structure or remotely through a web interface. To accomplish this, the visualization subsystem must interface with the automation and sensing subsystems to aggregate greenhouse performance and sensor data. After a discussion with the end users, the following requirements were determined:

- create a web application that hosts a general user interface for students and the public to view the current status of the greenhouse and historical data;
- create a database to store greenhouse sensor data and weather forecast data;
- provide a password protected page for administrators to control the automation settings;
- provide a control panel inside the greenhouse for changing automation settings and controlling the fans, lights, vents and water pumps;
- host the web application using an online hosting company; and

- create an intuitive user interface that is usable on a wide range of devices, from desktop PCs to mobile devices.

To collect and store greenhouse sensor data, present that data to users via an intuitive user interface, and to provide a control interface for the greenhouse a Java web application was developed. The web application was developed using a model-view-controller (MVC) architecture. It provides two different types of Java servlets, view servlets and API servlets. The view servlets provide the greenhouse user interface screens to end users connecting via a standard web browser. View servlets were created for a *Home* page, a *Settings* page, a *Sensor Data* page, and a *History* page. When queried, the API servlets return JSON objects that can be used to populate the user interface views or for controlling the greenhouse. The API servlets are also used to post new sensor data to a MySQL database. The database also contains a schedule that can be configured by a greenhouse administrator. The schedule allows an administrator to set specific times for watering, turning on/off lights, etc. The schedule is used by the web application to generate control signals that are transmitted to a control panel in the greenhouse allowing for remote control of the greenhouse. The Java web application is deployed using a Glassfish server. The web application and the MySQL database are both hosted using Amazon Web Services (AWS). Fig. 3 shows a diagram of the web application.

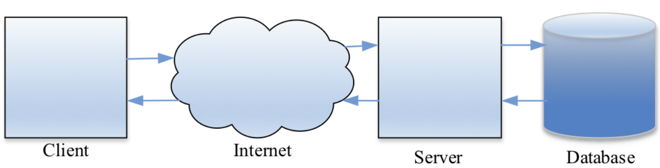

Fig. 3. Web application diagram

The main control panel for the greenhouse consists of a 10-inch touchscreen connected to a Raspberry Pi. The Raspberry Pi serves as a central point of communication and control for the greenhouse. It receives sensor data from the various sensor enclosures, transmits that data to both the backend web application and the automation system, and drives the touchscreen control interface. The homepage for the control interface, shown in Fig. 4, features an interactive 3D model of the greenhouse and the current state of the greenhouse. The left side of the interface displays a navigation panel that provides links to greenhouse settings, current sensor data, and historical sensor data. Fig. 5 shows the sensor data page in more detail. The sensor data page provides light, temperature, humidity, and soil moisture information for each of the six greenhouse zones. This user interface is accessible via the touchscreen on the main control panel located in the greenhouse or remotely using a standard web browser.

The Raspberry Pi in the main control panel also receives control signals from the backend web application that can subsequently be passed on to the automation system to control greenhouse functions. This is useful, for example, when a greenhouse administrator wants to manually turn off water, or turn on heaters remotely. The administrator can do so via the web application from anywhere and the web application will transmit those settings to the greenhouse. The backend web application also sends control signals to the Raspberry Pi in response to scheduled events that are stored in the backend database.

Fig. 4. Greenhouse homepage

Fig. 5. Sensor data for each zone

Manual control of the greenhouse is available on the settings page of the user interface. This page is reserved for administrators and is password protected. An authorized administrator can manually turn on/off fans, shades, lights, and water as well as set the temperature range and light threshold. Each of the six automation zones in the greenhouse can be individually scheduled or manually controlled by an administrator.

Historical greenhouse data is available via a history page on the user interface. This page provides users with multiple graphs displaying greenhouse sensor data for temperature, humidity, light and soil moisture values. Sensor data is displayed in 15 minute increments. Graphs can be generated for each of the six greenhouse zones individually or using average values for the entire greenhouse.

C. Power and Automation Subsystem

The automation system was designed to allow both manual and automated control of the greenhouse. Manual control may be necessary in the event that hardware failure occurs or simply to provide a manual override to the automated control system. The automation system enforces a hierarchy in its

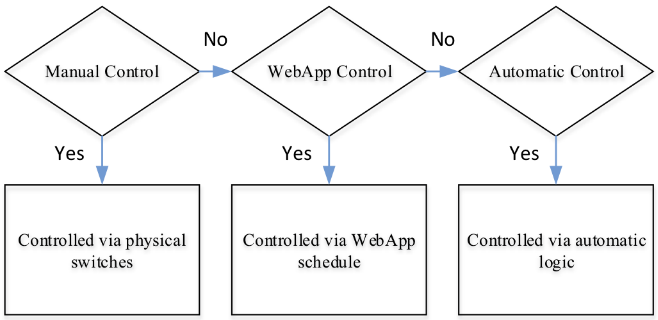

Fig. 6. Automation control architecture hierarchy

control architecture. This hierarchy is shown in Fig. 6. The physical manual controls of the greenhouse have the highest precedence; should a user set the heating to manual control, the automation and web application can no longer control the heating. When a user sets the heating back to automated controls the web application and automation logic can resume control. The web application is next in the control hierarchy. Similar to the physical manual control, the web application provides manual controls that can be used to turn greenhouse systems on/off at the discretion of an administrative user. Scheduled events, unless overridden by an administrative user, also run at this level of the hierarchy. The last layer of control is the automated control system. The automation software receives sensor data, web application control information, and greenhouse limits/schedules from the web application database, and controls the greenhouse systems accordingly.

The automation control hierarchy was implemented using five main components: a Raspberry Pi to perform all data processing and decision making, two relay boards for controlling power to all greenhouse hardware, an Arduino Mega to control those relays, and a circuit to allow for switching between manual and automated controls. The automation software running on the Raspberry Pi was implemented as a set of state machines, one each for light, temperature, water, and shade control. A diagram of the state machine for controlling temperature is shown in Fig. 7 as an example.

The control software was written in C#. Since, C# is not native to the Raspberry Pi, the Linux plugin Mono is used to run the .NET framework required by the automated control software. The choice to use C# was influenced mostly by the comfort in the team's ability to program in C#, rather than Java, C++, or other languages that are more readily supported by the Raspberry Pi. C# allowed for a high-level object-oriented approach to programming the automated control system. Additionally, libraries for JSON and HTTP were available and helped to streamline the development of the automated control system.

IV. Integration

The sensing, visualization, and automation subsystems are integrated together over an Ethernet network. The moisture

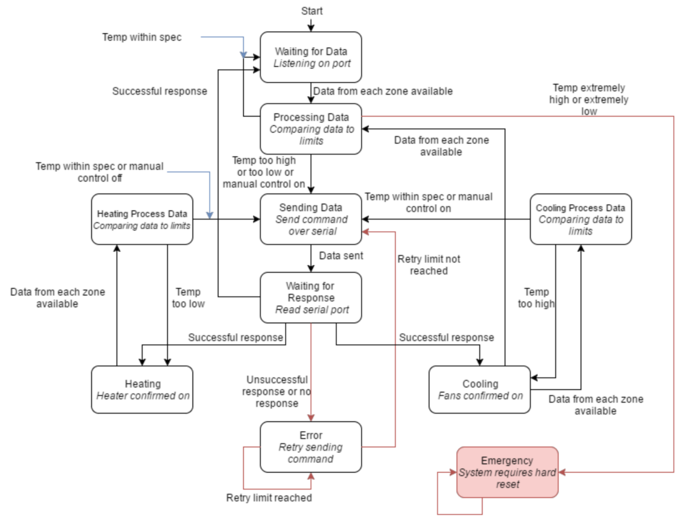

Fig. 7. State machine for temperature control

and TLH enclosures from the sensing system transmit sensor data to the visualization subsystem as UDP datagrams over Ethernet. Likewise, the power and automation subsystem communicates with the visualization team over Ethernet. The automation subsystem receives control signals from the visualization team to turn on/off greenhouse hardware systems. It also transmits state information back to the visualization subsystem to ensure the user interface is consistent with the state of the physical hardware systems. The automation subsystem Raspberry Pi integrates with an Arduino Mega to control the physical hardware in the greenhouse. An electric box is used to house all the electronics components and control circuits for the automated greenhouse. The Raspberry Pi for the visualization subsystem communicates with the backend web application over the internet. Fig. 8 shows the communication between subsystems.

V. Conclusions

The smart sensor network for the automated urban greenhouse was designed and implemented as a part of the Capstone

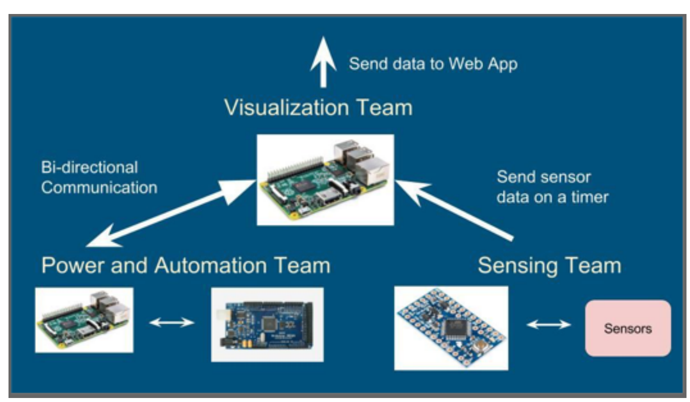

Fig. 8. Sensor network communication flow

Design course. During the two-semester design, build, and test process students successfully integrated a smart sensor network with a web application and an automated control system for the greenhouse. The smart sensor network incorporates precise sensing of temperature, light, humidity, and moisture. The web application allows for both on-site and remote control of the greenhouse via a standard web browser. The automated control system turns on/off greenhouse hardware in response to scheduled events or sensor readings. This project provided students a window of opportunity to solve a real-world problem as well as help a community in need. Fig 9 shows a prototype of the students' work.

Fig. 9. Greenhouse prototype testing

REFERENCES

[1] Economic Research Service, "United States Department of Agriculture," May 2017. [Online]. Available: https://www.ers.usda.gov/data-products/food-access-research-atlas/go-to-the-atlas/

[2] K. S. Kannan and G. Thilagavathi, "Online Farming Based on Embedded Systems and Wireless Sensor Networks," in *2013 International Conference on Computation of Power Energy Information and Communication (ICCPEIC)*, April 2013, pp. 71–74.

[3] P. C. Satpute and D. P. Theng, "Intellectual Climate System for Monitoring Industrial Environment," in *2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT)*, April 2013, pp. 36–39.

[4] K. P. Ferentinos, N. Katsoulas, A. Tzounis, T. Bartzanas, and C. Kittas, "Wireless Sensor Networks for Greenhouse Climate and Plant Condition Assessment," *Biosystem Engineering*, vol. 153, pp. 70–81, 2017.

[5] Adafruit, "Adafruit TSL2591 High Dynamic Range Digital Light Sensor," November 2017. [Online]. Available: https://www.adafruit.com/product/1980

[6] Adafruit, "AM2302 (wired DHT22) temperature-humidity sensor," December 2017. [Online]. Available: https://www.adafruit.com/product/393

[7] Meter Group, "Meter Environment," December 2017. [Online]. Available: https://www.metergroup.com/environment/