Smart Hive

Project Website: pwestman.github.io

Declaration of Joint Authorship

Roberto Loja, Yuri Sentsiv, and Paul Westman, the members of team Smart Hive, confirm that the concepts presented in this report have resulted from original thinking and research. All references to previously existing work have been appropriately cited throughout this document, as well as comprehensively accounted in its bibliography. As far as possible, we have evenly divided all work amongst ourselves. Though all members participated in all tasks to some extent, our primary areas of responsibility were as follows. Roberto Loja was responsible for interfacing our IoT hive with our remote database, by writing software that aggregates and uploads all sensor data into a format usable by client applications. He was also responsible for the strain gauge circuit used to gather weight data from the hive, as well as the design of the physical package of the Smart Hive hardware. Finally, Roberto contributed a significant portion of the code base for our mobile application. Yuri Sentsiv was responsible for the interface between our remote database and our mobile application, as well as designing and implementing the hardware system that, using temperature sensors, tracks the physical location of the bee cluster inside a hive. Further, Yuri was integral to the user interface design and usability testing of the mobile application, as well as the integration of the GPS system of target mobile devices. Paul Westman was responsible for developing the hardware and software for tracking a hive's ingress and egress, thus providing an indirect population count. Paul was crucial in ensuring the mobile application's compliance with Material Design guidelines, as well as designing the database schema and maintaining data consistency. Finally, Paul acted as the scrum master in the development of the application, coordinating the team's workflow, managing supplies, ordering parts, scheduling meetings, and keeping track of the project schedule.

Approved Proposal

Proposal for the development of Smart Hive

Prepared by Roberto Loja, Yurii Sentsiv, Paul Westman
Computer Engineering Technology Students

January 20, 2017

Executive Summary

As students in the Computer Engineering Technology program, We will be integrating the knowledge and skills we have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include a custom PCB with sensors and actuators for tracking and recording the movement of bees in and out of the hive. The database will store the population of bees inside the hive as well as temperature and humidity readings. The mobile device functionality will include requesting the most recent readings of population of bees in the hive, temperature and humidity and will be further detailed in the mobile application proposal. We will be collaborating with the following company/department: Humber Honey Bees.

Background

The problem solved by the project is finding a non-invasive way of tracking bee populations in the hive with varying temperature and humidity. With the depleted population of Honey bees recently, accurate data in this area is crucial. Up to date data can be requested and viewed from a mobile application that will be developed and integrated with the hardware component over the next two semesters.

The Humber Honey Bees are an initiative undertaken by Humber in June 2015 in an attempt to rebuild the local population of Honey bees in the area around Humber College. Honey bees are an essential part of our world as they are responsible for pollinating many of the plants that we eat. Due to their declining populations, studying and tracking them has never been more important. Therefore, this project will attempt to compile crucial data on Honey bee movement and population in the hive in varying temperatures and humidity.

We have searched for prior art via Humber's IEEE subscription selecting "My Subscribed Content" and have found and read articles that provide technical background information:

The first article provides insight into a smart bee hive that measures population, honey production, and temperature/humidity. (Wallich, 2011)

The next article introduces the use of strain gauges and instrumentation amplifiers. (Ştefănescu, 2011)

The last article demonstrates a method for estimating the population of a bee hive by measuring the hive's capacitance. (Perrault & Teachman, 2016)

In the Computer Engineering Technology program we have learned about the following topics from the respective relevant courses:

- Java Docs from CENG 212 Programming Techniques In Java,
- Construction of circuits from CENG 215 Digital And Interfacing Systems,
- Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,
- Micro computing from CENG 252 Embedded Systems,
- SQL from CENG 254 Database With Java,
- Web access of databases from CENG 256 Internet Scripting; and,
- Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable us to build the subsystems and integrate them together as our capstone project.

Methodology

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall and winter semesters. Our coursework will focus on the first two of the 3 phases of this project:
Phase 1 Hardware build.
Phase 2 System integration.
Phase 3 Demonstration to future employers.

Phase 1 Hardware build

The hardware build was completed in the fall term. It fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that was allowed to be used was 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC. Maximum power consumption was to be no more than 20 Watts.

Phase 2 System integration

The system integration will be completed in the winter term.

Phase 3 Demonstration to future employers

This project will showcase the knowledge and skills that we have learned to potential employers.

The tables below provide rough effort and non-labour estimates respectively for each phase.

Labour Estimates	Hrs	Notes
Phase 1		

Writing proposal.	9	Tech identification quiz.
Creating project schedule. Initial project team meeting.	9	Proposal due.
Creating budget. Status Meeting.	9	Project Schedule due.
Acquiring components and writing progress report.	9	Budget due.
Mechanical assembly and writing progress report. Status Meeting.	9	Progress Report due (components acquired milestone).
PCB fabrication.	9	Progress Report due (Mechanical Assembly milestone).
Interface wiring, Placard design, Status Meeting.	9	PCB Due (power up milestone).
Preparing for demonstration.	9	Placard due.
Writing progress report and demonstrating project.	9	Progress Report due (Demonstrations at Open House Saturday, November 7, 2015 from 10 a.m. - 2 p.m.).
Editing build video.	9	Peer grading of demonstrations due.
Incorporation of feedback from demonstration and writing progress report. Status Meeting.	9	30 second build video due.
Practice presentations	9	Progress Report due.
1st round of Presentations, Collaborators present.	9	Presentation PowerPoint file due.
2nd round of Presentations	9	Build instructions up due.
Project videos, Status Meeting.	9	30 second script due.
Phase 1 Total	**135**	
Phase 2		
Meet with collaborators	9	Status Meeting
Initial integration.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Meet with collaborators	9	Status Meeting
Incorporation of feedback.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Prepare for demonstration.	9	Progress Report
Complete presentation.	9	Demonstration at Open House Saturday, April 9, 2016 10 a.m. to 2 p.m.
Complete final report. 1st round of Presentations.	9	Presentation PowerPoint file due.
Write video script. 2nd round of Presentations, delivery of project.	9	Final written report including final budget and record of expenditures, covering both this semester and the previous semester.
Project videos.	9	Video script due
Phase 2 Total	**135**	
Phase 3		
Interviews	TBD	
Phase 3 Total	**TBD**	
Material Estimates	**Cost**	**Notes**
Phase 1		
Raspberry Pi 3 Model B	$80.00	Creatron Inc.
Peripherals with cables	$5.00	
Digital Bathroom Scale	$25.00	
Resistors	$2.00	
Infrared Optical Interrupter Module	$80.80	
DHT11	$17.00	

Phase 1 Total	**$209.80**
Phase 2	
Materials to improve functionality, fit, and finish of project.	
Phase 2 Total	**TBD**
Phase 3	
Off campus colocation	<$100.00
Shipping	*TBD*
Tax	*TBD*
Duty	*TBD*
Phase 3 Total	**TBD**

Concluding remarks

This proposal presents a plan for providing an IoT solution for bee tracking at Humber College. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating our ability to learn how to support projects. We request approval of this project.

Abstract

The western honey bee's prolific pollination of crops makes an enormous contribution to agriculture. As this has been threatened by colony collapse disorder, the time demands on experienced bee keepers have greatly increased. We have aimed to lessen those demands by allowing beekeepers to remotely monitor the health of hives. This project makes use of a compact package of carefully chosen sensors, mounted on a beehive, that wouldn't interfere with bee colony life, whose readings can be monitored from an internet connected tablet or phone using an app of our design. By lowering the time required to monitor each hive, we aim to allow beekeepers to tend to a greater number of hives, and to focus on those that require more attention, hopefully lessening the wider effects of colony collapse.

Illustrations and Diagrams

Table of Contents

1. Product Introduction

The decline of the bee population has been a major threat to the future of agriculture due to humans' reliance on their activity for pollinating crops. As technology has advanced, beekeepers have been left behind with very few effective options for monitoring the activities of the bees in a hive. This means that beekeepers do not have the most up to date data on the activity of the hive, such as weight, population, and cluster location, unless they are physically by the hive. Smart Hive aims to tackle this problem by incorporating sensors into the hive to track these metrics and upload them to a database that can be queried in real time by beekeepers using our Smart Hive application for Android mobile devices, and provide the most up to date information 24/7.

Sensors incorporated into the hive are powered by a Raspberry Pi to collect data on the population, weight, and cluster location of bees in the hive. This data is uploaded to a database and available for beekeepers immediately to make decisions on how to intervene to ensure the health of the hive. It is possible to monitor multiple hives within the application, displaying their status at the touch of a button.

2. Software Requirements Specifications

2.1 Product Description

2.1.1 Problem To Be Solved

This project aims to solve the problem of not being able to see inside a beehive to determine its overall health. By incorporating sensors into a beehive, beekeepers are able to get a deeper understanding of what is going on inside the hive and if intervention is necessary by the beekeeper in order to maintain the hive's functionality.

2.1.2 Intended Users

This product is intended for beekeepers who are looking for a way to more closely monitor what is going on inside the hives that they are responsible for.

2.1.3 Overview Of Product

Smart Hive includes a Raspberry Pi 3 Model B, as well as DHT11 sensors, Infrared Optical Interrupter sensors, and a wheat stone bridge for measuring the temperature, humidity, population, and weight of the hives.

2.2 System Description

2.2.1 Product Perspective

This product is open source, with the hopes that users will modify and distribute their own customized versions for the advancement of beekeeping metrics.

2.2.2 Design Constraints

Smart Hive is meant to operate year round to gather metrics on the hive's health. This allows beekeepers to determine if human intervention is required for the survival of the hive. However, the product is designed so that it will not impact the daily movement of the bees.

2.2.3 Product Functions

The sensors attached to the Raspberry Pi 3 collect data from the various sensors to provide a deeper understanding of what is going on inside the hive. The metrics that this product measures are temperature, humidity, population, and weight. The temperature and humidity give the beekeeper an idea of the climate that their hives are currently in. The population lets the beekeeper know if bees are dying and allows them to respond accordingly. The weight gives an idea of how much honey is stored in the hive at any time and if it will be sufficient to get the colony through the winter.

2.2.4 User Characteristics

The end user of this product will be a certified beekeeper who is actively managing one or more hives. The user must have an Android smartphone in order to monitor the status of the hive through the Smart Hive mobile application.

2.2.5 Constraints, Assumptions, and Dependencies

The mobile application runs on Android API 19 or higher. It works on a mobile phone or a tablet. The software that is running on the Raspberry Pi is in a Linux environment. The user of the application must be a certified beekeeper who is actively managing bee hives.

2.3 Specific Requirements

2.3.1 Database

Smart Hive uses Google's Firebase Database, which pairs easily with the Android application that was developed for users to view hive information. Firebase has enhanced statistics, authentication, as well as being a commonly used database for Android IDE. We have fairly straightforward structure, which will contain the user's unique ID that will be automatically created when the user is authenticated with the Google account. Under the user's ID will be entries for different hives, depending how many were activated. Under each hive, there will be information about the location, creation date, name, humidity, population, temperature, weight and date of last update. Further, the results will be fetched and displayed in the Android app. Roberto will be responsible for writing a short script that will execute once the Raspberry Pi has booted that will make a new node in the database associated with the user that is using it.

2.3.2 Web Interface

One of the functions of Google's Firebase service is providing statistics of app usage by end-users. All of the data in the database is presented in spanning list topology and it is relatively easy to track needed information. As the web interface is already set up, Paul will be responsible for maintaining the web interface and ensuring newly set up hives are being properly added to the Firebase console database.

2.3.3 Mobile Application

The Android application was developed in order to give users easy access to the up-to-date data on the hives. As stated previously, we are using Google's Firebase services, which includes database visualization as well as Google authentication that is used on the mobile application's Welcome screen. This allows users to login to their account and see their specific hive information and status. After the user has logged in, the main screen will display all of the hives in the list. From there user can click on any of the hives and see information about them. This information is fetched from the Firebase database. Also, the user can discover the hive on the map, which can be updated from the app in case the hive was moved. The information will be displayed in different colors, indicating if there are any possible problems. Yurii will be responsible for testing the mobile application once it is integrated with the database. He will also be responsible for deployment and subsequent version control the final version of the application (for example managing the Google Play Store page).

2.3.4 Hardware

The main processing unit that is used to collect the data and then upload it to database is the Raspberry Pi 3 microcomputer. The product uses 3 different sensors to track temperature, humidity, population, and weight. The DHT11 is used in different locations in the hive so the readings of temperature and humidity are more precise. Another function is locating the bee cluster, which is dependent on temperature readings in different spots in the hive. The Infrared Optical Interrupter module detects bees as they break the infrared beam between the two barriers on the sensor. With a sensor on either side of the entrance,it can track whether a bee is entering or exiting the hive and increment or decrement a counter accordingly. A wheatstone bridge is used from a modified bathroom scale and placed underneath the hive to measure they weight changes of the hive. The weight sensor will help monitor the honey production to ensure there is adequate supply to last the winter. Each of us will be responsible for writing threaded scripts that will gather the readings from the sensors that each of us are responsible for: Roberto is responsible for weight data, Yurii is responsible for cluster location data, and Paul is responsible for population count.

2.3.5 Performance

All of the changes that are made are displayed in real time in the app as long as the user has an internet connection. The data retrieved by the hardware becomes more useful over time. This is because patterns associated with population and temperature become clearer with longer periods of time. But overall, if the system is running for a few days, the population readings that are uploaded to database should be more precise than when it was just launched.

2.3.6 Functional Requirements

The functional requirement for Smart Hive include being able to update the data in the database in real time and be presented on the mobile application. Therefore, there should be constant connection to the internet on the Raspberry Pi. Smart Hive can use the Ethernet connection or Wi-Fi, which are both built into the Raspberry Pi 3. There should also be a constant power supply to the Raspberry Pi so the system can constantly retrieve and update the readings.

2.4 Additional Requirements

2.4.1 Security

The Smart Hive mobile application uses Google authentication to verify the user before they are able to access any of the hives data. If the user does not have a Google account, one can be created from the main page of the application.

2.4.2 Safety

Beekeepers should always wear industry standard protective equipment when physically interacting with any active bee hives. This is necessary for the initial setup of the hardware on the beehive.

3. Conclusions

4. Recommendations

5. Progress Reports

6. References

Perrault, P., & Teachman, M. (2016). Bee counters: Measuring a nest's occupation by its capacitance [resources_Hands on]. *IEEE Spectrum*, *53*(2), 20–21. https://doi.org/10.1109/MSPEC.2016.7419791

Ştefănescu, D. M. (2011). Strain gauges and wheatstone bridges #x2014; basic instrumentation and new applications for electrical measurement of non-electrical quantities. In *Eighth international multi-conference on systems, signals devices* (pp. 1–5). https://doi.org/10.1109/SSD.2011.5767428

Wallich, P. (2011). Beehackers [hands on]. *IEEE Spectrum*, *48*(5), 20–21. https://doi.org/10.1109/MSPEC.2011.5753235