Plan Bee

Tariq Ausaf, Giovanny Reyes, Yannick Roberts, Katelyn Winters

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This project is focused on the creation of an internet-of-things, sub-1ghz communications template and retrofitting this with sensors for monitoring the health and outputs of remote, commercial beehives. This project will cover mems and sensors technology to achieve tracking of metrics; this will include but not be limited to RF and integration of a GPS locator, load cells, a thermistor, a humidity sensor, accelerometers and vibration sensors. The hardware technologies will be weatherproofed for placement upon the commercial beehives. IBM Watson will be discussed and used for intelligent algorithms, and amazon AWS will be shown used for the storage and manipulation of the apiary's data. Other web technologies will be shown used to create an application "dashboard" for the commercial apiary.**

***Index Terms*— Internet of Things, RF signals, Solar Panels, Sensor Arrays.**

I. Introduction

The global market of connected devices is estimated to exceed $7.1 trillion by 2020. As such, the demand for a general-purpose asset tracking solution may be significant enough for an entirely new market segment to emerge. A cost-effective device capable of gathering data from a series of connected sensors and broadcasting this information to the cloud for further analysis. Our technology-in-development is aimed at accomplishing this task, generally appropriate for most Internet-of-Things application. Our technology is a general Commercial Sub-1Ghz Communications Template. We will focus on implementation details required to develop our technology, with the capability of funneling sensory data across a secured internet connection to a cloud connected database for analysis and logistical planning.

It is worth noting that we are developing this for a specific application because it fits our senior design project needs and allows us a budget, but this is not our final intent. Our final intent is a versatile Internet-of-Things (IoT) application communications template that can be applied directly to several applications for asset tracking, such as transcontinental transportation of goods, service applications such as agriculture farms, as well as many others. These are applications that we intend to market this device toward following senior design and intend to develop an end-product for. Implementation details required to create this sensory-communications device are presented here in an easy-to-follow manner aimed at a general engineering-experienced audience. However, those with non-technical background, with some thinking, should be able to follow our logic.

It is understood that bees are a vital component of the ecosystem. Many flowering, food-producing crops are reliant upon bees for reproduction through pollination. Honeybees are responsible for pollinating greater than thirty-percent of food crops grown in the United States each year. Honeybees are productive members of the agriculture ecosystem. As such, Beekeepers – Apiarists – are of vital importance to the agriculture economy, and thus the national security of the United States as well. We were linked to this industry through a personal connection with Steven Eisele, owner of the commercial apiary company Pollination U.S.A.

Commercial honeybees, which accomplish the bulk of the pollination of U.S. agricultural farms, are facing problems in today's changing climate. Like humans, the health of a beehive factors into account many different individual bees – like cells – that work best unanimously when healthy, and at the mercy of disease and pests. Over the past several years, on average, beehives have been dying at a rate of about thirty-percent-per-year. The many issues that can affect bee hives and how this project implements a versatile Internet-of-Things (IoT) application communications template that can be applied directly to several applications for asset tracking will be examined below.

II. Issues Examined

A. *Hive Health Issues*

The loss of so many honey bees can be attributed to many things and at times only one small incident can be very devastating to an entire apiary system. There are times when an entire colony can be lost, and this is known as Colony Collapse disorder which can happen in the span of less than 24 hours. This is largely due to several health concerns beehives face: pests including varroa mites and small hive beetles, viral and fungal infections, and invaders including other aggressive, territorial beehives, and Africanized bees. There are different signs that can alert apiaries if there is something wrong in a certain colony and/or hive ranging from the weight of the hive, to even the temperature of the hive and we intend on calculating these factors in an effective manner.

B. *Location Issues*

When apiaries rent out their pollination bees to farmers, these farms are normally in very far away, remote areas. The hives are kept in yard-arrangements, and yards are kept distanced from each other Five-thousand individual hives are usually spread across one-hundred-square-miles. And these hives are all remotely located, and usually in extreme environments, making maintenance and upkeep of every individual hive unfeasible economically. This is also a good reason why this project should also rely on solar panels as its power source since it would be inconvenient and economically unfeasible to have it run any other way.

III. Targeted Measurements and Requirements

When considering our requirements, we wanted to make sure that our goals and the market goals of the consumer went hand in hand, and for this we took the advice of Steven Eisele so that we could properly match both engineering and market requirements.

In terms of the communication, one main aspect was event reporting of important data and the latency of said data. We wanted to send the required data from the hives to the user as quickly as possible, and we also needed it to be able to cover at least 10 miles in order to have proper, real-time communication. The device itself also needed to be weather proof since it would be outside exposed to the elements, and it should be able to fit a certain dimension of 8” x 8” x 6”. Another important consideration was that the device should have a good peripheral scalability; that is, that the device should be able to be customized to hold even more sensors in the future to suit the individual needs of apiarists. The project also needed to be properly affordable for it to be used, and this also required that the device be effectively powered via solar power.

As for the information we tried to gather Steven Eisele, our sponsor, noted that he would like to see a system for each individual beehive health monitoring, for reasons stated above, that keeps track of the following:

A. Weight

Hive weight and its rate-of-change is the single most important statistic to track for commercial apiaries. Constant weight tracking gives the beekeeper information on when the bees are leaving (at dawn) and resting (at night) in the hive, how many bees are active within the hive, and the amassing of honey within the hive until the S-curve levels, when it is time to harvest. Weight monitoring is one of the simpler hive monitoring techniques, however it also provides extensive data feedback throughout the year. Measuring the weight of a hive over time can show whether the hive honey output is progressing or declining. This can give valuable insight on the health of that specific hive. Smaller weight changes each day can show when the bees come and go from pollination, or if there are any abnormalities with their behavior in general. Not only can weight measurement give details about the bees in a hive, but it can also provide information on what type of plants the bees are pollinating and when they flower. This is all obtained by weighing each hive individually for an extensive period; when the weights are applied, they do not have to be operated from that point onward, meaning they are very easy to maintain.

Project Bee found a way to utilize only one load cell instead of a standard four load-cell structure at each corner. Normally four load cells are required for extreme precision, but since we only need to see a drastic change in weight then only a single load cell was required. We designed and implemented a small load cell board that we then could connect directly and communicate to our overall PCB.

B. Location of Bee Hives (GPS)

During transportation to contract location and during contract duration, live location of the physical beehive (GPS) needs to be tracked by both the beekeeper and customer. This is important because of the high frequency with which these beehives are relocated, during which they are often lost or stolen. Precise location tracking is one proposed solution to this problem.

C. Temperature and Humidity

Honeybees optimally need to maintain a brood (childhood) nesting temperature between 32 – 35 degrees Celsius for growth. Temperature fluctuation of about 15 degrees in either direction from these temperatures is acceptable, because of mechanisms bees have developed to regulate hive temperature. Without brooding, it is acceptable for temperatures to range from about 0 – 50 degrees Celsius. Similarly, relative humidity between 50 – 60% is optimal for brood nesting, while outside of brooding through 20% - 100% relative humidity is acceptable. During brooding then, it is especially important to track temperature and humidity within the hive.

D. Forceful Hive Movement

Being out in the wilderness, the hives can be exposed to different dangers such rummaging animals like bears in search of honey, or forceful winds like those experienced in Florida during hurricane season. This is why it is important to be able to measure if the set dimensions of the hive are being changed in real time.

IV. SYSTEM COMPONENTS AND DESIGN

Project Bee will be focused on the hardware portion, as defined in the illustration below. Accordingly, a private server will be used to emulate the entire software portion. This includes the web application aggregator and data management unit. Various cloud-based tools will be researched for their feasibility, given the time constraint required to complete the design. While a PCB will be developed for the hardware tracking device, the WAN Gateway will be made an Amazon AWS instance. The gateway will also use wired ethernet to communicate with the internet. In an ideal scenario, a gateway may feature numerous ways of connecting to the internet, notably, cellular, Wi-Fi, satellite or one of the many IOT standards being deployed.

When thinking about the proper components needed to create this project what came to mind right away was that overall, we needed to have an appropriate microcontroller. Our microcontroller needed to have the functionalities that could take in many sensors as well as meet the needed RF communication requirements. Once this was chosen, all other components could be easily chosen to revolve around the microcontroller's main setup.

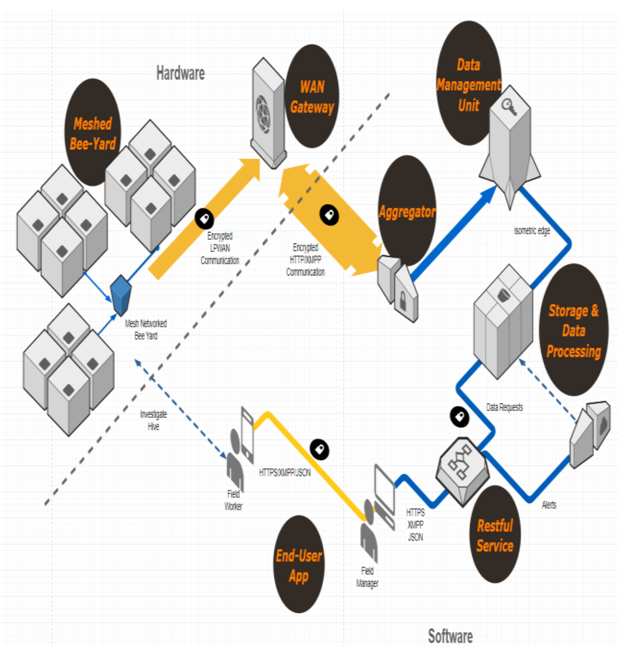

Figure1: Overall Network Design

A. Microcontroller

When considering our microcontroller, we researched many but always looking for it to ultimately have the goal of being able to have different sensor sources as well as an ample RF communication setting. We also desired an MCU that could be not only easily programmable via XDS but could also access Sub-1GHz technology in order to create our desired network of communication between hives and devices.

In the end, we decided to use the Texas Instrument CC1352R. The reason we picked this is because of its wide range of use. It has an ARM Cortex M4F Processor, an RF Core Processor, and an ARM Cortex M0 Embedded Processor. This allows it to support UART, both I2C and SPI, and carries 28 additional GPIO (both analog and digital). This way we can not only implement a wide variety of sensors with different programming settings, but it can also help support both long-range and short-range RF communications.

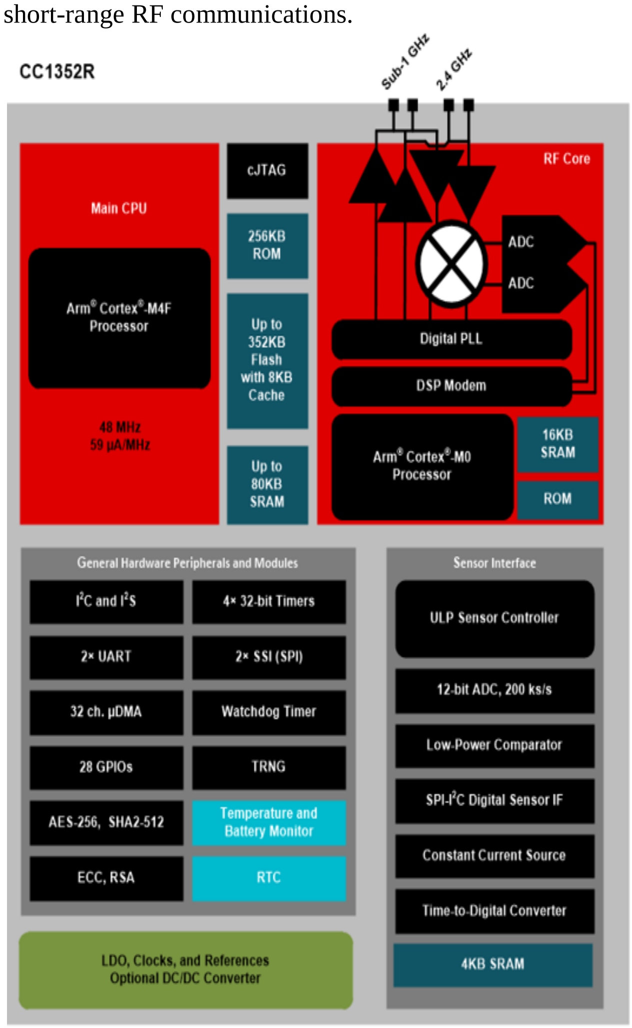

Figure2: MCU Layout

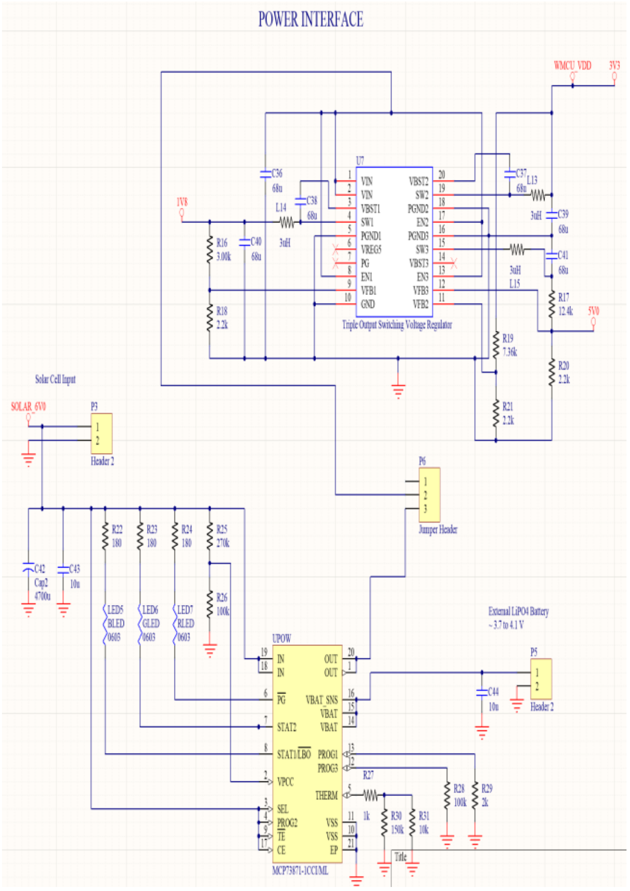

Figure3: Power Schematic

B. Power System

To utilize the photovoltaic cells to power our project, we will first run the solar panel's output voltage through a rechargeable battery that will then deliver the voltage gathered to the rest of the PCB. Certain parts of this battery system require control such as making the solar panels stop charging the battery when it is fully charged as well as measuring how much charge is left in the battery for the recharge cycle to be triggered to restart. In order to power this project, we needed to pick components that would help implement a low-cost, solar power energy input that would be properly measured for our components. Luckily, this device is very low-power from the get-go which means we didn't have to worry too much about certain under-powering and energy loss issues.

For the solar panel, we chose a simple 6 Volt photo-voltaic cell and we linked this to the MCP73871 solar charger. This charger is special because it does not do True Power-Point (MPPT) tracking which is preferable for charging LiPo batteries as the one that we are intending to use. MPPT is normally preferred in order for the solar panel to output the correct amount of voltage versus current but this is not needed in this case. This solar charger was chosen because it helps connect the solar panel to the device and the battery while also charging the battery at the same time. It will also always pick to draw power from the solar panel first before drawing from the battery.

When the power is being pulled it then passes through a voltage regulator which we picked to be the TPS6580. This is used to step down the voltage from the rechargeable battery and from the solar panel that is originally too high for the PCB. It provides a DC to DC transformation from the power that must be picked up at a moderate level from the solar panels to the much smaller values required to power our board effectively. For this project we only needed to step the voltage down and not up as we only needed to use 3V3 for nearly all of our components with this controller also giving the options of utilizing 1V8 and 5V0. Another advantage to this regulator is that it is a switching regulator instead of a linear regulator which makes it more efficient for solar panel projects.

C. Sensors

For measuring physical movement of each individual bee hive we picked the MPU9250 Accelerometer. This part was mainly chosen due to its low cost in contrast to the others researched. This sensor is an accelerometer and gyroscope. It is a 3-axis sensor that uses these 3-axis to detect change in positioning in any of these 3 to send out a specific signal. This sensor can be programmed through both I2C and SPI protocols, has a simple operating voltage of 2.4-3.6 V, and can operate at a temperature range of -40˚C to 85˚C

We also implemented a pressure sensor in order to measure atmospheric pressure since bees have a hard time flying at low pressure and it hinders the feeding process and overall honey production. The main characteristics of this sensor came down to it being a pressure sensor that operates under 4 Volts was prioritized due to the power source providing around 4V or less to the system. I2C was also seen as a necessity but the chosen one could have different output protocols. Although a high maximum pressure is not needed, the highest degree of accuracy is crucial for this project. In the end, the selected chip was LPS22HBTR and this was due to its accuracy of ±.015 PSI as well as fitting below the 4V threshold. This was also the cheapest sensor found.

For another important consideration that could affect the health of the colonies, we looked into how much ambient light was entering the hives as this can normally be a sign of certain hive infrastructure failures. For this we chose the OPT3001 ambient light sensor. This sensor outputted I2C only. Something worthy of note for this sensor is that taking special care to place surrounding components that fall into this category either above the sensor or away from the sensor is of upmost importance because small amounts of ambient or emitted LED light can seriously affect the readings of this sensor.

For the humidity and moisture sensor we used the HDC2010. This sensor was also an I2C output and this was suggested to be kept apart from most other components in the PCB so as for internal temperatures to not affect the readings. This sensor was introduced because the humidity inside the hives can be a sign of the viscosity of the honey being produced which is a sign of good honey, which in exchange is the sign of a good and healthy hive.

D. Communication and GPS

Our main communication is the Sub-1 GHz technology in order to achieve long-range communication. To accomplish this, we used a monopolar antenna. We were able to apply a simple antenna without impedance matching because of the fact that all of our components that we gathered (including the antenna) were all kept at a 50 Ohm basis.

Because we intend to use multiple channels, an RF switch is needed; a MUX cannot be used because of the high impedance, and digital signals will not be used. RF switches route high frequency signals through transmission paths by cleverly utilizing the signal frequency as the controller for output, in the same way the MUX uses binary logic for control. Because we will be using two frequencies for inputs and outputs, we will need a transfer switch for double pole, double throw (DPDT) which allows inputs from one channel and outputs to two channels, and likewise the reverse is true for signal intake.

Another key component to our antenna implementation is that the antenna uses a differential input for both its negative and positive signals, so it is necessary to use a balun for single line integration.

For our GPS we went with a Radionova_M10578-A2. This chip had similar protocols to the MCU as well as running on 50-Ohm routing lines and its data is outputted through UART, so it was easy to integrate with the MCU.

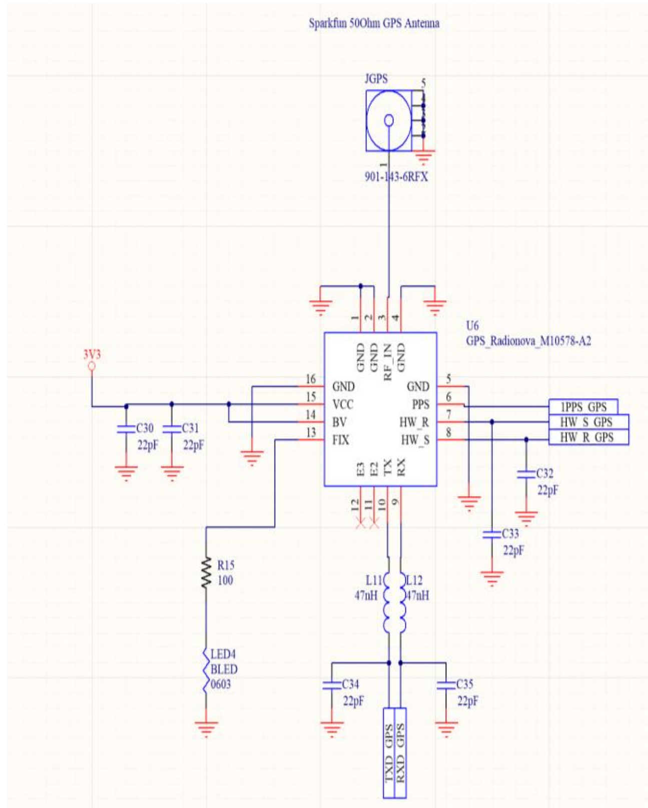

Figure 4: GPS Schematic

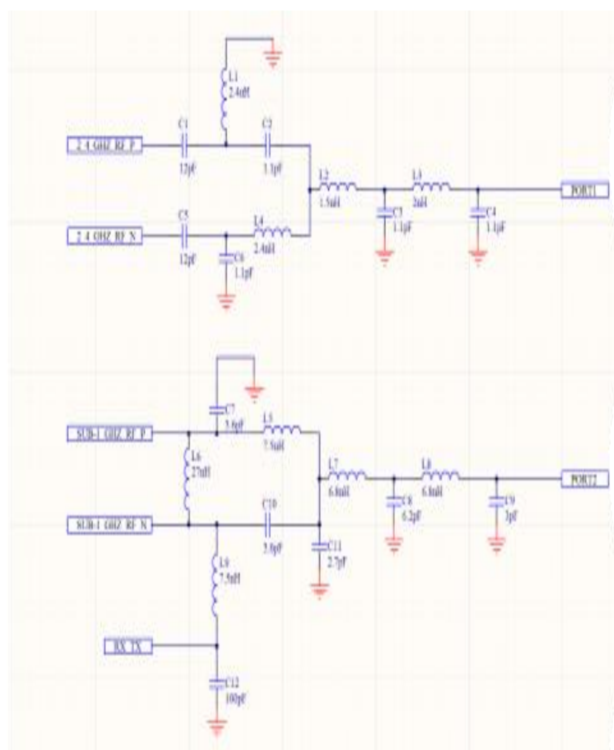

Figure 5: RF Balun Setup Schematic

V. SOFTWARE

This section attempts to describe the overall application integration and application flow details. The software portion of Plan-Bee will be divided into 5 stages. Each stage is assigned a range of responsibilities which will enable the project to accomplish the listed requirement specifications. The project's primary goal would be to present sensor data, collected from some arbitrary environment, to an end user. Each stage plays an integral role in accomplishing such task, and hence cannot be decoupled, or isolated from the rest of the toolchain.

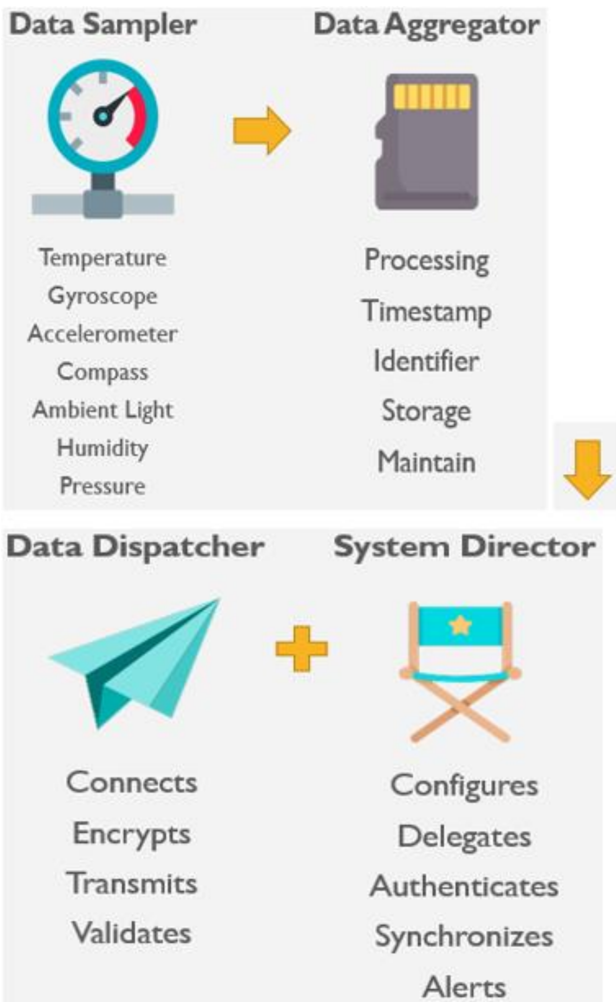

Figure 6: Software Path

A. Sensor Controller

This application is responsible for gathering sensor data and pushing said data to the main controller unit. The application is executed on the low-powered ARM processor dedicated to reading sensory data and consists of multiple tasks being executed in a periodic manner. As a result, the overall periodic invocation of the application will be dependent on total sensory tasks to be executed. Each task is responsible for initializing a communication device, initializing the sensor device, reading sensor data, and terminating both sensor device and communication device. When the sensor data is read, it is then stored in main memory for the data logger to store the required data to non-volatile storage.

B. Data Dispatcher

The Data Dispatcher is executed on the main controller unit (MCU). It is responsible for both logging and dispatching data and, therefore, comprises for two tasks, the Data Aggregator and Data Dispatcher. The data aggregator stage is usually invoked by the Sensor Controller. Upon invocation, the Data Aggregator reads the raw sensor data; does the necessary translation, depending on type; appends additional data elements to aid in the querying process; and finally stores the resulting data structure in non-volatile memory. The Data Dispatcher complements the Aggregator by scheduling what types are to be transmitted to an access point, and transmitting said data while also managing routing and other transmission details

C. Forwarding Gateway

The Forwarding Gateway lives on the access point. This application is responsible for translating sensor data to valid values to be stored on a local database, and synchronizing said database with a registered cloud server or using a connected mobile device. Because of time constraints the synchronization process will encompass copying the database elements to an attached device and have the device manage synchronization. To keep transfers small, successfully synchronized data will then be archived and removed for an active syncing database. The Forwarding Gateway may also act as a quasi-application server: providing data to an authenticated mobile device. For a stretch goal, if the access point is unable to connect to the internet, the mobile application me be used as a network gateway to synchronize data with the cloud. This may permit users to continuously synchronize local data with the cloud, even in remote areas without wireless internet access.

D. Web Service

This application is responsible for receiving sensor data and presenting said data to an authenticated user through request or alerts. Sensor data could be received from an internet connected access point or from a mobile device. Upon receiving specific data, the Web Service will inspect the data for abnormalities. If found, the Web Service will trigger an alert for an authenticated end user. The Web Service will also provide a robust API for accessing said data. This will allow both online applications and mobile applications to access and present the data using the most effective user interface for the device in use. We use Amazon AWS for data storage to a database and pushing the data to the end user application.

E. User Application

The end User Application will be executed either as a mobile application or a web application. The main responsibility for a user application is to obtain data from either a Web Service application or Forwarding Gateway application and present it to the end user in a intuitive fashion for the end user. Th e end user application should also be able to synchronize its local database with that of the Web Service. This will allow data aggregated to a gateway to be pushed to a web server through the end user's mobile device.

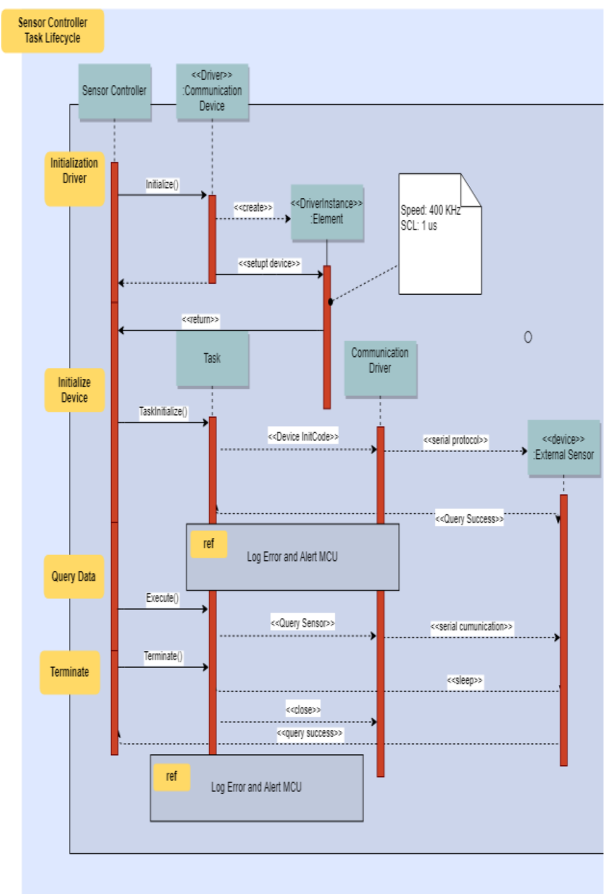

Figure7: Data Sampling Sequencing Diagram

V. PCB DESIGN

For our PCB we took it upon ourselves to create the most efficient use of space while keeping in mind certain design characteristics needed for RF transmission. In the end, we soldered everything ourselves which was quite time consuming. We kept our design to a four-layer layout: one for RF Component Routing (GPS and CC1352) with 15-mil thick dielectric, one for power also with a 15-mil thick dielectric, one for routing everything else, and one for round with a 50-mil dielectric between it and the power plane. We put the GPS module on the top-right, about 20mm away from CC1352R1. The balun and Impedance Matching were embedded already. Our normal routing, we kept to 7 mils thick through the board. The user interface was put top-center, the on-board sensors bottom center, and the headers and peripherals, XDS debugging, and power were all put the on far left.

In order to protect our PCB since it will be constantly outside and subjected to the elements as well as the possibility of wild animals we designed and 3-D printed a hub to protect the main PCB while still leaving holes to plug wires for power and off-board sensors.

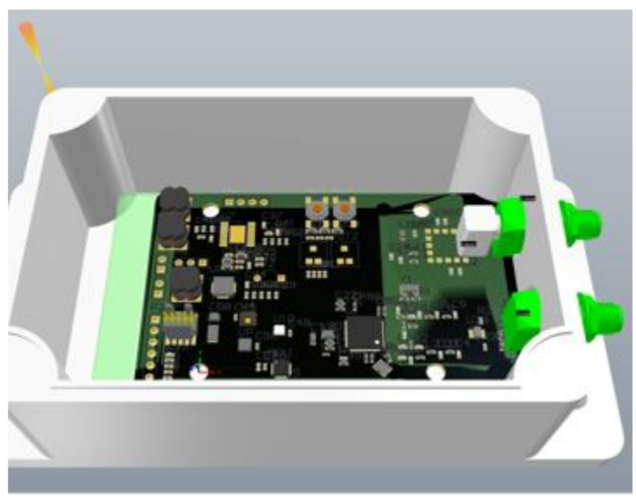

Figure 8: PCB and Housing

VI. TESTING

A. MCU/Sensor Testing

Initially, Texas Instruments CC1100 was used to testing low powered wide area network range. This was done using a custom application built by a company called Anaren. This application can send temperature data across 2 MSP430 using a Texas Instrument CC1100 wireless transceiver. While not as powerful as Texas Instrument's

CC1352R used as the main MCU of this project, the CC1100 does feature TI's long-range communication stack. The CC1100 development boards were flash with Anarens default application and asserted to be able to wirelessly communicate at short distances. One of the boards was then moved until communication halted. A range of approximately 50 ft was also recorded. We happily sided with Texas Instruments CC1352R due to the additional gain promoted. Two development boards were purchased, and one attached to a breadboard in order to test Texas Instruments' sensor hub booster pack. This booster pack houses numerous sensor devices and is easily interfaced with the CC1352R development board using the I2C port. The following section goes into detail on the testing methodologies. Sensors utilizing I2C were purposefully chosen primarily for the ease of use and expandability. I2C provides a robust method of communicating with many devices and allows additional devices to be incorporated in a preexisting design. Such an architecture allows the primary microcontroller to communicate with as much as 128 different devices. The Sensor Hub Booster Pack, designed by Texas Instruments, was used to test I2C communication and sensory data acquisition. The Sensor Hub features an MPU9150 by Invensense; this device integrates a 3-axis gyro, 3-axis accelerometer, 3-axis compass and a temperature sensor in to a single device

B. RF Testing

To rapidly prototype wireless communication, Texas Instruments' SmartRF software tools was used to configure the RF transceiver to communication across various parameters and transmit an arbitrary set of data across differing ranges. SmartRF allows rapid application testing and can provide additional features such as wireless sniffing and wireless integration techniques using various wireless standards such as Bluetooth and Zigbee. To begin, a CC1352R development board was attached to a computer using USB. The devices were then activated in the SmartRF software and its features queried for transferring data. SmartRF offers a set of configuration options which presents an assorted amount of communication method to be used for communication. The SimpleLink long range communication configuration was then used to test the long-range communication.

A variable length data packed comprised of randomly generated data was then created. The packet consisted of an initial 16-bit preamble followed by a 32-bit sync word. This is then followed by an 8-bit length value, then an 8-bit address value and finally the variable length data. Another CC1352R development board was attached to a second computer and the receive option was selected in SmartRF studio. This selection option was matched with that of the transferring board and the reception was initiated. The reception device presented the packet data as generated by the transferring device.

To test overall reception distance, the transfer method was set to continuous and the device on the receiving end was relocated. Given the transferring device was elevated 12 feet above ground, an estimated communication distance of about 1500 feet was achieved. While this distance was significantly less than what is proposed, we do believe the distance could be improved by installing a better antenna and utilizing more power. In addition, the test took place in a moving vehicle within a rural environment. Both of which would have an adverse effect on the final operating distance. Further testing will be done to achieve a greater distance. Texas Instrument has also promoted their CC1352P; this chip features a higher power wireless transceiver and is also pin compatible with currently selected CC1352R. While the chip is not yet available; when it is, I do believe migration will be drastically simplified because of application abstraction made possible with Texas Instruments' RTOS.

VI. CONCLUSION

The research and development of Plan Bee was extensive and often took certain backwards turns away from our goal, but we were able to create the most appropriate design possible in order to meet our selected goals. Our team believes that not only will this product create a lasting, positive impact on the future of the pollination industry, but can also be a stepping stone into a larger field of data gathering and asset tracking for many agricultural and commercial settings. Since this project can be easily amplified, the Plan Bee team will look into improving our design our performance.

BIOGRAPHY

The Plan Bee team was comprised of four electrical engineers who researched and partook in all fields of this project equally, yet had some at certain times some more specific roles: Tariq Ausaf who specialized in RF design, Giovanny Reyes who specialized in powering systems and some communications, Yannick Roberts who focused on al software design, and Katelyn Winters who specialized in PCB design and sensor research/design.