THE RANCH HAND

Final Report

Charles Anderson
Randy Ardywibowo
Jeff Bartlett
Connor Furqueron

Sponsored By
William Goh and
Texas Instruments

We would like to say a special note of thanks to William Goh and the team at Texas Instruments for their generous sponsorship of our project.

Change Record

Rev.	Date	Originator	Approvals	Description
A	9 / 19 / 2016	Connor Furqueron Charles Anderson Randy Ardywibowo Jeff Bartlett		Draft Release
B	10/10/ 2016	Jeff Bartlett		
C	11/21/2016	Jeff Bartlett		
D	12/04/2016	Randy Ardywibowo		Final Report
E	04/29/2017	Randy Ardywibowo		2nd Semester Final Report

TABLE OF CONTENTS

THE RANCH HAND

Charles Anderson

Randy Ardywibowo

Jeff Bartlett

Connor Furqueron

Concept of Operations

CONCEPT OF OPERATIONS

1 EXECUTIVE SUMMARY

The agriculture industry currently lacks cost effective and practical means to monitor the condition of their livestock across great distances and large grazing areas. This results in low yields and increased effort on the rancher. Current methods of livestock include GPS tracking and RFID wands. The first method can be costly and the second is limited to tracking tagged animals within a specified perimeter. We propose to solve this problem with an IOT mesh network based RSSI tracking system. The Ranch Hand tracks the location of cattle by using a sensor tagging system to detect the location of cattle in the field. The location data is then transmitted through a mesh network to a base station, allowing the system to operate without adding costly base stations in the field. The system is powered through a system of photovoltaic panels and batteries, eliminating the need for manual charging by ranchers and allowing the system to operate autonomously with little maintenance.

2 INTRODUCTION

In the agriculture industry, determining the location of livestock congregation is important for several reasons. Tracking livestock location allows ranchers to find missing livestock that has wandered away from the herd. Livestock positioning can determine how much livestock to place in individual fields, notifying ranchers when to rotate their fields to prevent erosion and pollution. Tracking is also vital to maintaining livestock safety. In 2010, more than 15,000 livestock were stolen, incurring a loss of 9.3 million dollars (USDA, 2011). When taking a look at other main causes for cattle and calf loss, it makes sense to monitor and keep track of livestock [1].

Number of Head and Total Value of Cattle and Calf Death Loss by Cause – United States: 2010

[Totals may not add due to rounding]

Cause	Number of head	Percent of total	Total value
	(number)	(percent)	(1,000 dollars)
Predator			
Coyotes	116,700	53.1	48,185
Mountain lions and bobcats [1]	18,900	8.6	9,221
Dogs	21,800	9.9	10,067
Vultures	11,900	5.4	4,641
Wolves	8,100	3.7	3,646
Bears	2,800	1.3	1,415
Other predators	12,400	5.6	6,352
Unknown predators	27,300	12.4	14,948
Total predator	219,900	100.0	98,475
Non-predator			
Digestive problems	505,000	13.4	267,799
Respiratory problems	1,055,000	28.0	643,146
Metabolic problems	59,800	1.6	47,558
Mastitis	62,000	1.6	59,112
Lameness/injury	140,900	3.7	112,251
Other diseases	179,500	4.8	114,577
Weather related	489,000	13.0	274,092
Calving problems	494,000	13.1	274,670
Poisoning	36,100	0.9	26,817
Theft	15,100	0.4	9,309
Other non-predator	301,600	8.0	247,092
Unknown non-predator	435,000	11.5	276,476
Total non-predator	3,773,000	100.0	2,352,899
United States Total [2]	3,992,900	100.0	2,451,374

[1] Includes cougars, pumas and lynx

Table 1:Livestock deaths in 2010

Currently, ranchers lack cost effective means to monitor the location of livestock across great distances and large grazing areas. Current methods of tracking livestock are short range RFID tags and longer range GPS tracking. Short Range RFID tracking can help to identify an animal, but requires a close-up encounter and can be time and resource intensive. GPS tracking attempts to remedy this but can be costly and often must be maintained by a 3rd party contractor. This cost can be limiting to smaller farms looking to tag and track all of their livestock.

From this problem statement, it is clear that one optimal solution would be a system that is capable of tracking the location of livestock with a cheaper and more accurate network. Several challenges arise in creating such a solution. The solution must work in all weather conditions, constraining the type of sensor suitable for this application. The system must also be easy to install and maintain, calling for self-sustaining energy and low-power solutions. Finally, the solution must also be cost effective, able to be installed without producing a substantial financial burden on the owner.

Weighing these factors, we propose to create a mesh network sensor system. Tagging livestock with radio frequency inspired tags with built in transmitters, the system can track the distance of livestock to fixed transceivers spread across the field. The position of the cattle can then be estimated through triangulation. This position information is then relayed to other transceivers using a mesh network configuration to a base station and into the rancher's computer workstation. The rancher's computer will then analyze the data and display it in a graphical user interface. The system will be solar powered, allowing it to operate autonomously with little maintenance.

2.1 Background

Several sensors have been used in detecting and locating living organisms. These sensors include Radar, Sonar, LIDAR, Low-light Camera, and Thermal Imaging. Arguably, these sensors are more robust in tracking, as tracking isn't limited to tagged objects and subject behavioral data can be inferred from these sensors. However, due to the significant price of these sensors, the best method for our application is using triangulation with RFID tags.

Another prospective class of tracking sensors are proximity sensors. These sensors detect movement around a specified area by their proximity to the sensor. This may be used to detect the location of livestock, however, the sensor may be prone to noise as there are many potential moving objects in a field.

RFIDs are commonly used in position location and triangulation of tagged objects. Two types of RFIDs can be used in this method, passive RFIDs and active RFIDs. Passive RFIDs draw no power, however, they have a limited range of detection. On the other hand, active RFIDs draw some amount of power, but they have a relatively large range of detection. Existing livestock tracking methods commonly use close range RFID or GPS long range triangulation. However, these methods do not incorporate mesh networks which cause them to draw a significant amount of power or can be resource intensive to monitor.

Solar power is one method to allow systems to harvest naturally ambient energy. In utilizing solar power, the system takes advantage of solar energy, which is abundant within an open field during daytime. Additionally, by using solar energy, one could extrapolate the time that the system could operate autonomously in remote locations with minimal user maintenance. Used in conjunction with a bank of batteries, the system will be able to operate at night. Normally, for electronics application, solar power is gathered using photovoltaic cells organized into arrays and panels, and this solar power system can charge the battery bank to capacity during daytime.

The Ranch Hand uses a mesh sensor network backed by Texas Instruments CC1310 processors. Because of their range and processing power, the sensors are able to provide efficient and reliable tagging locations. Texas Instrument's MCUs are propelled by a Contiki mesh network standard, which uses efficient routing protocols to quickly contact the home sensor for data relay. The Ranch Hand's network requires less sensors and can handle a greater distance between sensors, which cuts costs and complexity.

Our livestock detection method can be used beyond its intended usage scenario as a cattle tracking system. The system can be utilized to track almost anything that is tagged within the system's area of operation. This allows the system to be used to track other organisms or objects such as pets or objects within a facility. Everyday consumers could track the location of a pet through a backyard or a child through a house. Large warehouses, shipping yards, or supply depots could tag boxes or vehicles and be able to locate any particular box or vehicle within the area of operation, which would have applications for large distribution companies and militaries.

2.2 Overview

2.2.1 Livestock Detection and Sensing

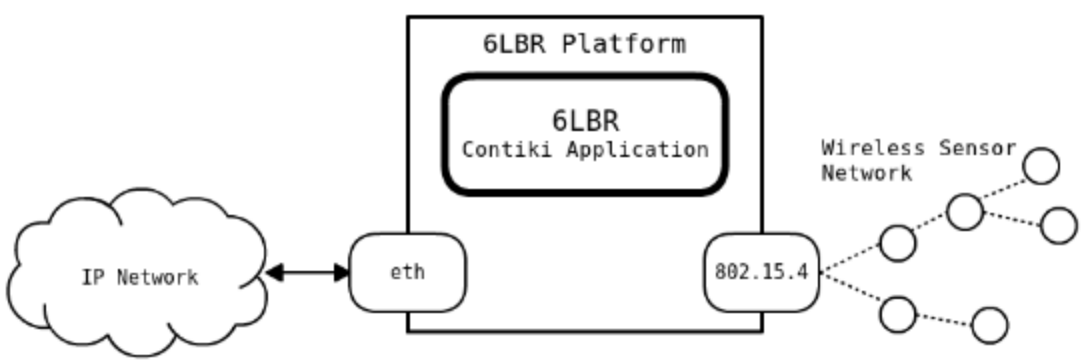
Livestock tracking has many practical applications to farming and research. Whether for security, identification, or analysis of any herding or behavioral patterns, the ability to track tagged assets reliably is vital. Detection for the Ranch Hand is based on tags that are both long range and low powered. These tags provide for fewer sensors and a wider area of mapping with the mesh network.

Tags use a low power consumption by 'pinging' an ultra-high frequency (UHF) signal to the Ranch Hand's sensors. Using a Contiki compatible transmitter, each tag is affordable, low power, and have a range that exceeds pricy existing tags.

These tags will broadcast to each field sensor stations, which are stationary field stations that can receive signals from the tags. The RSSI is recorded to be sent back through the mesh so the livestock can be tracked. This also requires the pre-registered GPS coordinate of each sensor be input.

2.2.2 Network and Data Transmission

Another innovative aspect of The Ranch Hand apart is its mesh network consisting of Texas Instrument's CC1310 backed sensors. Each sensor is programmed to quickly and effectively route tag information to the home station for processing using a 6lowpan mesh and a 6lbr border router. The standards of the system are built on 802.15.4 IEEE broadcast standards and is compatible with IPV6 networks.



Improving on sensor range is also an important feature of The Ranch Hand's sensors. With greater range capability for communication between sensors and tags, less sensors are needed. This helps to effectively cut costs and improve the communication of the system with fewer nodes for routing.

2.2.3 Analytics

Data from the mesh is sent over IP to the cloud server that is reserved for recording and modeling information from the field system. The data can then be displayed through the mapping API to so that those managing the livestock can have immediate access to their data.

2.2.4 Power

To power each sensor unit within the meshgrid network, each unit will have a self-contained power system. A static distribution of solar cells will be used to run electronics and charge a battery bank in the daytime within each substation. In nighttime conditions, the battery bank will power each sensor substation. This combination allows the system to take advantage of energy harvesting and create a much more autonomous and longer operating system. The active RFID tags will use low power batteries to allow them to operate for very long durations, up to a couple of years. Power will be supplied to the components attached to the 6lbr router by USB.

2.2.4.1 Photovoltaic Panels

Photovoltaic panels will be used to harvest solar energy from the open field and allow the system to operate with minimal maintenance for long periods of time. Using a static distribution of solar panels inhibits the efficiency of the system compared to panels that track the sun but greatly simplify the controls complexity of the power system. Photovoltaic panels deliver direct current power, which most of the system already utilizes; however, a charge controller and a system of disconnects is necessary to ensure proper voltage levels and proper flow of power through the system.

2.2.4.2 Batteries

A bank of batteries will be used to power the sensor stations at nighttime. This battery bank will be charged by the solar panels in the daytime. To ensure that power is properly delivered to the system, the same network of disconnects used for the photovoltaic cells will be employed. The batteries that will power the RFID tags will be small and low power, able to keep operating for long periods of time. These batteries, due to their long lifetime, will limit the maintenance required of the tags.

2.3 Referenced Documents and Standards

IEEE 802.15.4 is a standard which specifies the physical layer and media access control for low-rate wireless personal area networks (LR-WPANs).

IEEE C95.1-2005 Standard for Safety Levels with REspect to Human Exposure to Radio Frequency Electromagnetic Fields, 3kHZ to 300GHz.

Humane Handling of Livestock and Poultry (United States Department of Agriculture) regulates the handling of livestock.

IEEE SCC21 and the National Electric Code (NEC) regulate the use of renewable energy sources, photovoltaic cells, and many other aspects of the power system.

1184-2006 - IEEE Guide for Batteries for Uninterruptible Power Supply Systems regulates the choice of batteries for uninterruptible power supply systems.

3 OPERATING CONCEPT

3.1 Scope

The Ranch Hand system will be designed with tags attached to a desired target to be tracked. Using low-powered sensors arranged in a field, the tags will be detected by the sensors. The sensors then relay this information of a detected tag back to a base station by transmitting data from one sensor to another in the field through a mesh network approach. At the base station, the location of the tagged target will be able to be determined if three sensors have detected the tagged target through triangulation algorithms. By the completion of this project in May 2017, a subset model of the system will be constructed with five sensors that can relay data among one another, two tags that have a long battery life and emit identification data, and a base station computer program that will record and analyze data with a user interface to display the location of the tagged targets in the sensor grid. The model will be built with a budget of $500 from Texas A&M University and supplemented with the following integrated circuit components from the project sponsor, Texas Instruments: CC1350, Beaglebone Black, BQ25570.

Key features of the system that differentiate it from other tagging systems are its ability to work in all kinds of weather in the outdoors, having energy efficient tags using batteries, eco-friendly sensors using a mix of solar power and rechargeable batteries, and the ability to detect a variety of targets from animals, to people, to products and goods.

3.2 Operational Description and Constraints

The Ranch Hand will be primarily used by ranchers who wish to track the rough whereabouts of their cattle or sheep in an open field. The system will be set up by having numerous sensor stations spaced out in a specific order in the field to be monitored. The cattle or sheep that are to be tracked, will then be tagged with special, high-performance tags. The computer software of the system will then be installed on the rancher's personal laptop or computer. Through a simple user interface, the rancher will be able to utilize the computer software to observe the location of tagged cattle or sheep in the field.

Due to the client base being primarily ranchers, the budget of this target client is limited. Thus, the cost of the system will try to be minimized to account for the limited budget of a rancher clientele. The range of the system is also limited to the client's budget, due to increasing costs in adding more sensors to cover a larger field to monitor. Animal handling standards must also be followed when deciding on a method to tag the target. In this system, tags attached to the animal's ear will comply with modern animal handling standards.

3.3 System Description

In order to accomplish tracking of cattle through an open field, the system is composed of three different main components operating together to reconcile the location of cattle within a field. These three components are the tag, the sensor system, and the home station. A diagram of the system can be found in Figure 1

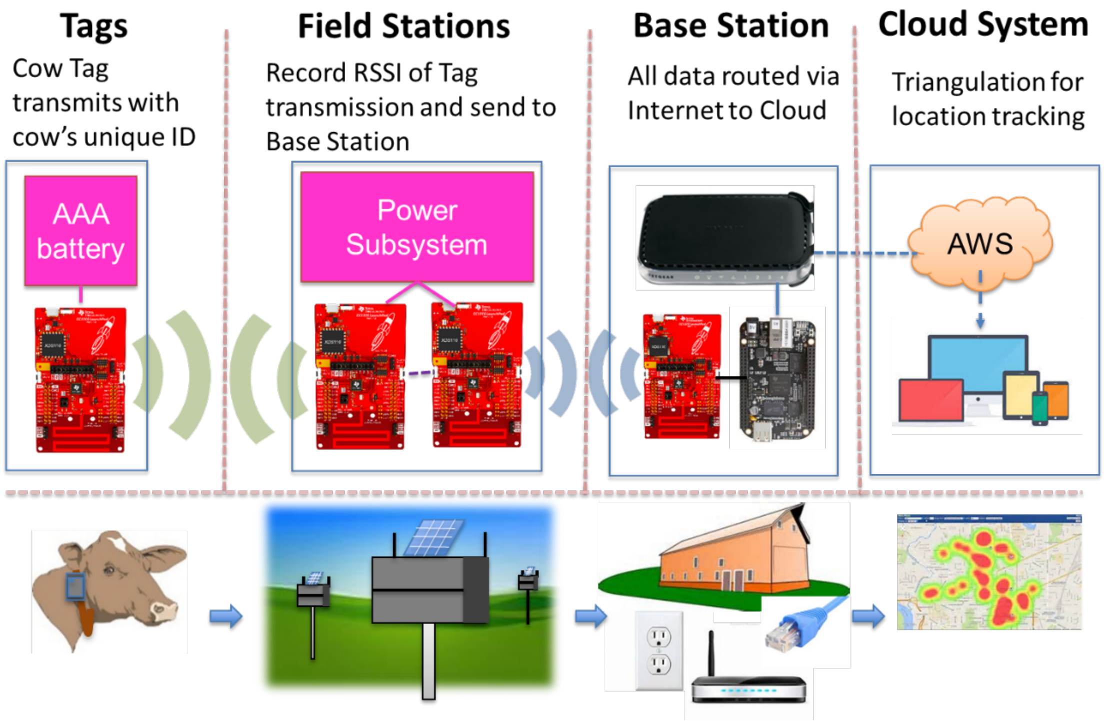


Figure 1: System Overview

The tag is an active RFID tag that transmits a signal out to the sensor systems at set time intervals. Placed through the ear of the livestock, the active RFID tag will allow the system to monitor the location of the cattle by RSSI triangulating the position of the RFID tag. Each tag will be powered through a battery, and because of the low power nature of each tag, the battery should be able to power the tag for at least a few months depending on ping frequency. Additionally, each tag will have a communications system composed of a transmitter and a control system that mediates the duration of time between successive transmissions.

The sensor system is composed of portable stations that are set up at particular locations within a field. Each station will pick up the signal from the active RFID tags placed on the cattle, and be able to calculate the RSSI of this transmission. The sensor system will then transmit this RSSI data through a mesh network built of the other stations back to the home station.

Each portable sensor station is composed of three subsystems: the field receiver subsystem, the mesh subsystem, and the power subsystem. The field reciever subsystem is composed of a receiver that detects the active RFID signals and a transceiver that allows data to be received, transmitted or relayed through the mesh network of stations. A CC1310 microcontroller with any peripheral packages comprise the receiving side. This will calculate the range of the signal intercepted from the receiver and coordinates the activities of the antennas. The CC1310 on the mesh side will receive this data over UART and send this via UDP to the home base. The power subsystem allows the stations to harvest energy in the form of solar power. Solar panels, placed in a static distribution, will be used to charge a battery bank and run the electronics of the other two subsystems in the daytime, while the battery bank will allow the system to operate at night. Additionally, within this power system, control and power regulation infrastructure, such as charge controllers and disconnects, will manage the power quality and flow of power.

Finally, the home station will receive ranging data from the mesh network and relay this data to a client PC, where triangulation is performed to determine the location of a tag within the geographical area of the field. Once this calculation is accomplished, it is sent to a graphical user interface to be easily seen by the rancher. Thus, there are three subsystem components: the communications components, the microcontroller component, and the cloud component. The communications component will be a transmitter that receives information from the mesh network of sensor stations, and relays this information to the microcontroller, a BeagleBone Black. The microcontroller will then aggregate and relay this information to the client cloud. The software component, installed on the client PC, will then use this data to calculate the position of the tag using triangulation and plot this calculated position to the graphical user interface, which makes the data easily accessible to the rancher.

In concert, these three system components (the tags, sensor stations, and home station) will allow the location of tagged cattle. Although the overall system is quite complicated, breaking it down into smaller functional blocks and components allows us to separate functionality and solidify the designs of simpler circuits and functions. See Figure 2, Figure 3, and Figure 4 for more detail of the steps.

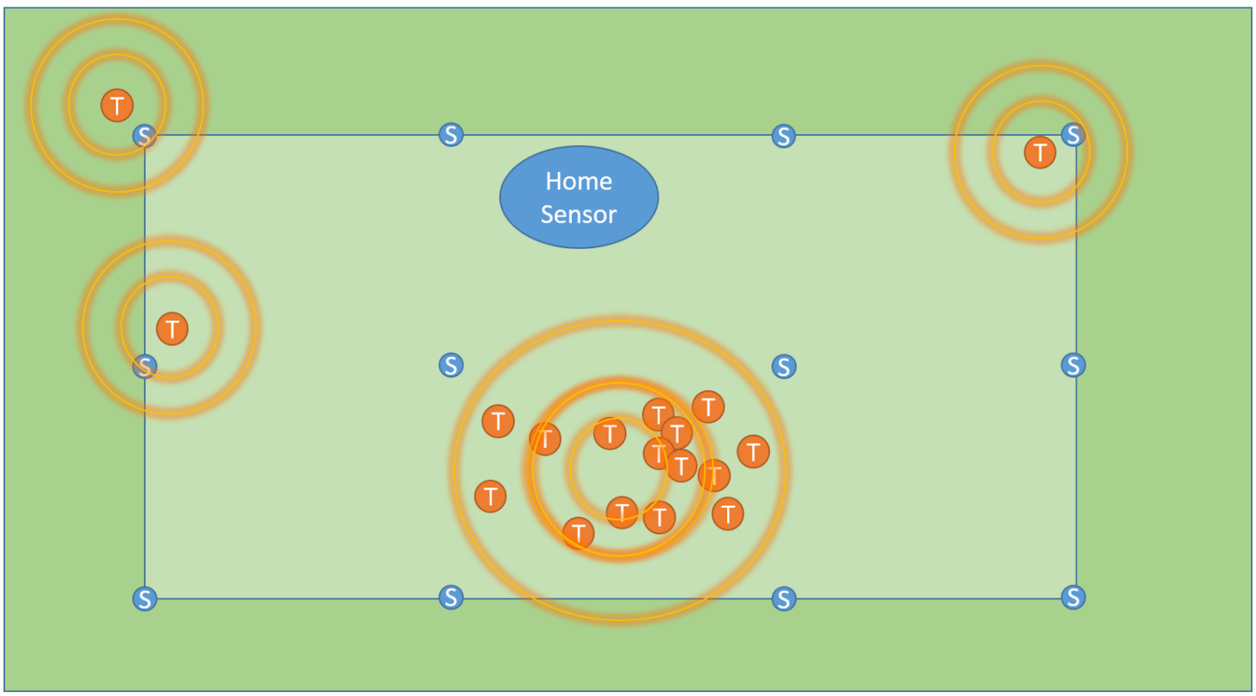


Figure 2: Tag Emission to sensors

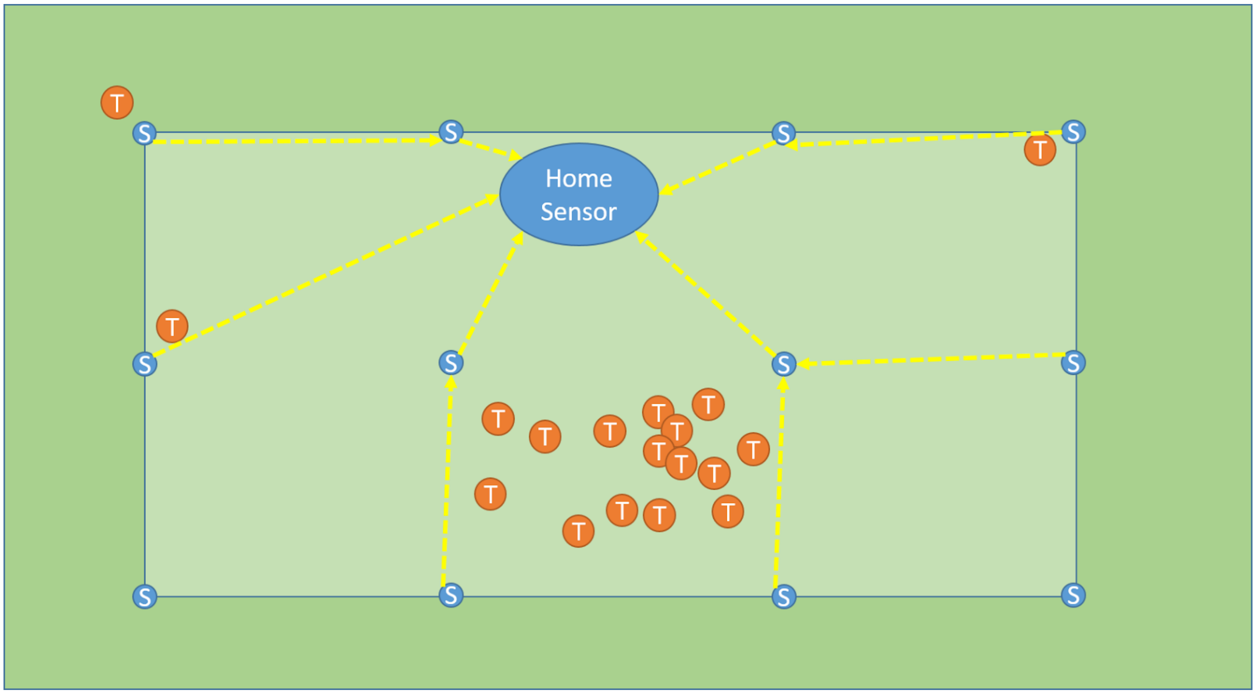


Figure 3: Mesh routing to base

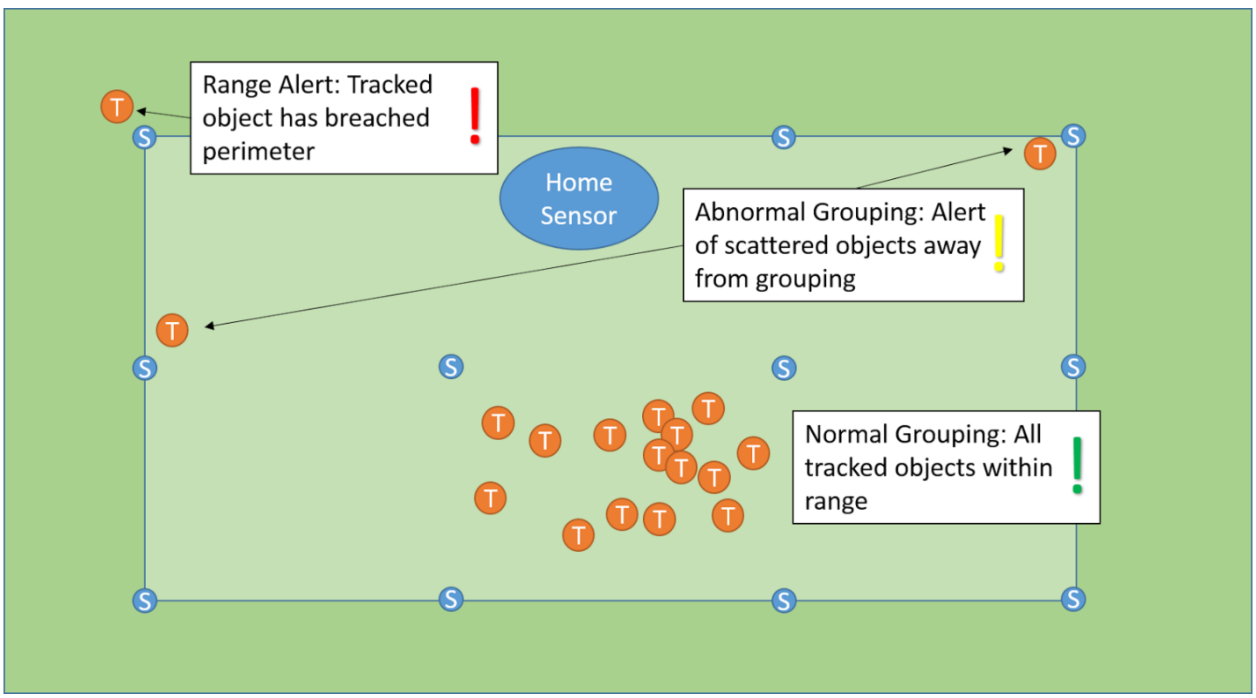


Figure 4: Analysis yields data for API

3.4 Modes of Operation

The system will be able to provide various modes of operation so that the end user can better achieve his or her goals. The modes of operation will be determined from the current algorithm being executed on the client PC system, so that the field-deployed components do not change in their operation. Essentially, each mode of operation is just another way to analyze the incoming data about the location of individual heads of cattle and the herd.

3.4.1 Basic Location Data Delivery

In its most basic mode of operation, the system will simply tell the rancher where exactly each tagged head of cattle is within the system's area of operation. It will also keep track of each livestock's tracked path within user specified timeline. This is an attractive feature for grazing and health maintenance purposes with livestock.

3.4.2 Perimeter Breach Detection and Location

Building upon basic location data delivery, the system could also monitor the edges of the system's area of operation for animals that have breached a safe perimeter defined by the rancher. This would simply require an algorithm be implemented in the client PC system to analyze if any tagged animal has gone beyond a particular boundary or has left the range of the system altogether. Additionally, the rancher would be able to see either the current location of the animal beyond the boundary, or if contact has been lost with the tag, the last known location of the animal. This mode of operation would aide the rancher in finding lost livestock and preventing livestock from escaping a pasture.

3.4.3 Herd Analytics

Another mode of operation which could be deployed to complement the perimeter breach detection would be to analyze the behavior of the herd within the system's area of operation. One could be able to detect aberrations within the herd. If animals have left the herd, an animal has not moved in an excessive amount of time, or any other aberrant behavior occurs, the system could send an alert to the rancher, allowing the rancher to go into the field and check the condition of the herd and the reason behind the aberrant behavior. This could allow the rancher to find injured or sick animals in time to allow medical aide to be given, or discover the death of an animal and move to prevent other deaths within the herd due to the same cause.

3.4.4 Other Modes of Operation

Because the modes of operation are based on algorithms implemented in the client PC software package, additional modes of operation could be invented that were not initially envisioned. This could lead to an ecosystem of various modes that would allow the system to metamorphosize to better meet the needs of the individual user. This exciting possibility would allow for solutions and analytics capabilities that could be developed and fleshed out that could expand the usefulness of the system and make it marketable to many more customers beyond the agricultural community. Additionally, the ease of implementing an alternative mode of operation would be as simple as installing another analytics algorithm within onto the client PC system.

3.5. Users

The nature of the system allows for some more uses beyond its initial conception as a cattle tracking system. Although its primary consumer base would be ranchers, the system and its technology could be utilized to track almost anything that is tagged within the system's area of operation. Thus, as long as the tag could be applied to a particular living thing or object, it could be tracked by the system. Everyday consumers could track the location of a pet through a backyard or a child through a house. Large warehouses, shipping yards, or supply depots could tag boxes or vehicles and be able to locate any particular box or vehicle within the area of operation, which would have applications for large distribution companies and militaries.

Because of such a large variety of potential customers, with a primary focus on the agricultural community, one must assume that the average customer would have limited technical knowledge and limited amount of time to learn to operate the system. In addition, we must assume that the requirements of customers would vary from one customer to the next, with uses and requirements for the system as varied as the users themselves. Also, the data collected by the system must be able to reach the user regardless of where the user is or where the system is located. This requires that the system be able to deliver the data reliably and operate with the systems already in place by the users. Also, the price point of the system must be appropriate for ranchers and average consumers to purchase. In the aggregate, these characteristics of the system users create interesting challenges that must be solved in order to make the system deployable to many different customers.

3.6. Support

The construction of the system will be simple enough that the user should be able to operate it effectively with minimal support or maintenance. Optimally, the system should be able to operate in remote locations, and with minimal setup overhead. Thus, the system should need very little support. However, some support must be provided to the customer to ensure successful system setup.

Therefore, support will be provided to potential customers in the form of training manuals. These training manuals must be produced for a nontechnical audience and be available in a large number of mediums to allow for the widest dissemination to users as possible. In addition to a printed manual, the manual should be provided in various electronic formats, able to be accessed from the internet. Additionally, due to the ubiquity of ranching and farming across humanity, one must be prepared to deliver this manual to customers in many different languages.

Additional support should be provided in case the customers need clarification of or have questions extending beyond the scope of the manual. There should be multiple avenues for customers to reach this support, including but not limited to a number that customers can call and an email address to which requests can be submitted.

4. Scenario(s)

4.1. Herd Dietary Habits Through a Pasture

This scenario outlines a common application of the Ranch Hand in agricultural practice.

A farmer has determined a few of her cattle have been showing intestinal distress. She decides to review the information that has been collected and analyzed from the Ranch Hand. Upon review, she notices that the sick cattle have all been grazing near the same spot by a watering hole. Her inspection yields a type of plant that can be harmful to her cattle, and she immediately sprays a pesticide to kill the plant. In the meantime, she brings the cattle to the other side of the farm and sets the Ranch Hand's alert system to inform her if any of her livestock go near the spot she sprayed.

5. Analysis

5.1. Summary of Proposed Improvements

Several improvements were identified over existing active RFID tagging systems. The proposed system does not need existing infrastructure, such as cell towers or satellites, in order to operate effectively. In fact, the only component that does not come with the system is the client PC. In addition, because the signals from the tags only need to be detected by the sensor stations, we can produce a much cheaper tag that can operate for longer periods of time. This is due to the fact that the electronics involved require much less power in order to contact the sensor stations. Also, because analytics are accomplished on the client PC via analytics algorithms, the user is able to customize the system to provide the most useful information.

5.2. Disadvantages and Limitations

There are some disadvantages and limitations that come along with our solution to tracking cattle. The first and greatest limitation of the system is that cattle must be tagged in order for the system to track them. This requires that the rancher be able to tag calves early from birth. Also, the entry of outside entities, such as thieves or predators, cannot be detected. Secondly, the performance of the analytics is grounded in the performance of the client PC. If the client PC is not robust enough, the analytics solutions may suffer inaccuracies or be delayed by the speed of onboard computation. This would become an especially pronounced issue as the herd of tagged cattle became larger. Perhaps another limitation is the lifetime of the tags. Eventually, the batteries in the tags will die, necessitating either the replacing of the batteries within tags or replacing the tags themselves.

5.3. Alternatives

While the Ranch Hand may not be able to track un-tagged targets, it makes up for this with an affordable and reliable tracking system that can be used every day. However, there are still many alternatives that could be pursued that could accomplish the objective of our solution. Some alternatives may be better than our solution for particular users and vice versa.

An interesting alternative that could be deployed would be to use thermal imagers to identify entities within the field. In using thermal imagers, one could detect other entities besides the cattle, such as thieves, predators, or other invading animals, and could also keep track of calves from birth. However, thermal imager are prohibitively expensive and performing image analysis in the sensor stations would require a significant amount of power and computation.

A second alternative would be to use a drone in combination with computer vision sensing technologies, or tag tracking antennas, to periodically map the field and locate cattle. This would eliminate the need for expensive multiple expensive thermal imagers if computer vision were chosen to be employed. However, the frequency of data input would be limited by the flight time and recharging period of the drone. Additionally, a massive amount of infrastructure and overhead within the project would be dedicated specifically to the development of the drone.

The Ranch Hand
Charles Anderson
Randy Ardywibowo
Jeff Bartlett
Connor Furqueron

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – Final
04/13/ 2016

FUNCTIONAL SYSTEM REQUIREMENTS

1 PURPOSE AND SCOPE

The purpose of this document is to delineate in measurable metrics the technical requirements present in our system. First any reference documents and other applicable material is presented. Then, the technical requirements of the system and existing between the subsystems are discussed. Finally, any support requirements are discussed. As a very general and broad functional description, our system is meant to track tagged targets through an area via radio ranging and triangulation and route received data to a base station through a mesh network.

2 RESPONSIBILITY AND CHANGE AUTHORITY

The team leader, Jeffrey Bartlett, has the responsibility to make sure the requirements of the system are met. William Gho, the representative of Texas Instruments, is the only one that can authorize changes to the system requirements.

3 APPLICABLE AND REFERENCE DOCUMENTS

3.5 Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Document Number	Revision/Release Date	Document Title
1	Ocotber, 2016	CC1310 SimpleLink™ Ultra-Low-Power Sub-1 GHz Wireless MCU
2	July, 2016	MSP432P401R, MSP432P401M Mixed-Signal Microcontrollers
3	March, 2015	bq25570 Nano Power Boost Charger and Buck Converter for Energy Harvester Powered Applications

3.6 Requirement Specific overview

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification, and are not controlled by their reference herein.

Document Number	Revision/Release Date	Document Title
4	2015	Energy Harvester Booster Pack

3.7 Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as "applicable" in this specification are incorporated as cited. All documents that are referred to within an applicable document are considered to be for guidance and information only, with the exception of ICDs that have their applicable documents considered to be incorporated as cited.

4 REQUIREMENTS

This section defines the minimum requirements that the development item(s) must meet. The requirements and constraints that apply to performance, design, interoperability, reliability, etc., of the system are covered.

4.1 System Definition

The system consists of 4 main subsystems: the tag system, the sensor system, the sensor mesh network, and the base station.

The tag subsystem are composed of the devices which are placed on the cattle. This subsystem is used to transmit radio frequencies (RF) to the sensor system such that each cattle's position can be detected. This system shall consist of an RF transmitter powered by a low power battery.

The sensor system is the subsystem that receives the RF frequencies from the tagging system. Several instances of this subsystem shall be scattered throughout the field to cover the entire herding area. This subsystem shall utilize an RF transceiver to both receive signals from the tag system, as well as send them to the sensor mesh network. The subsystem shall be solar powered and shall utilize a TI MSP432 microprocessor for onboard signal processing.

After the RF signal from the tag system has been received by the sensor system, the sensor mesh network relays this information to a base station. This mesh network shall be composed of transceivers from different sensor system across the field.

Finally, the cattle location information is received on a base microprocessor and processed on a computing workstation. The computing workstation shall process the information and display it on a user interface. The information shall consist of cattle location, herding tendency, general congregation area, and battery life of each cattle's tag system.

4.2 Characteristics

4.2.1 Functional/Performance Requirements

4.2.1.1 Tracking Accuracy

The positioning error of cattle shall not exceed 2 meters. This is the average length of a regularly sized cattle.

4.2.1.2 Minimum Area of Coverage

The system shall have a minimum area of coverage of three sensors of 300 square meters.

4.2.1.3 Lifetime of a System

The tag system shall have a battery life of no less than 2 years. The sensor system is not dependent on battery performance, and shall last the solar panels or the battery's charging cycle life span, whichever comes first.

4.2.1.4 Terrain

As uneven land elevation will distort triangulation calculations and barriers will weaken signal transmission, several terrain performance requirements need to be set. The system will function optimally on a ranch terrain with some barriers and minor elevation changes. The system will function optimally in terrain where the elevation does not vary more than plus or minus 5 meters and where there are more than 2-3 barriers per m^2.

4.2.1.5 System Capacity

There exist several different metrics for the capacity of the system. The system shall be able to track at least 24 tags simultaneously. The system shall be able to effectively route the information from the 24 tags at once. In addition, the system shall be able to operate effectively even when all 24 tags are together with a density of tags per square meter of 1 tag per 2 square meters, which is the approximate maximal density of cattle over an area.

4.2.2 Physical Characteristic

In the following sections, the physical characteristics and requirements of the system are discussed.

4.2.2.1 Dimensions

Tags shall have a weight not in excess of ¼ of a pound and not exceed physical dimensions of 2-½''x5''x1''. Sensor stations shall not have a weight in excess of 6 lbs and shall not exceed the physical dimensions of 14''x11''x5''.

Rationale: Tags should be small and unobtrusive to the cattle and the sensor stations must be light enough and have a small enough cross section to safely mount to the tops of poles, such as fence posts.

4.2.2.2 Range

Tags shall be detectable by the sensor stations up to 100 m away. Sensor stations should have a transmission and reception range of up to 1 km.

.

Rationale: Tags should be able to transmit an ample distance to the sensor since cows will not be guaranteed to remain near the sensors and several sensors must pick up the tag's signal in order to triangulate the position. Sensors should be able to transmit long distances so that if one section of the network fails, data will still be able to reach the base station.

4.2.3 Electrical Characteristics

4.2.3.1 Inputs

Each tag will be preprogrammed with a chip ID via USB on the CC1310 Launchpad. The CC1310 on the tags will receive a 900Mhz signal from the sensors. Inputs to the sensors will be a signal from either a tag or another sensor, along with the appropriate power for that sensor from the solar panel.

4.2.3.2 Power Consumption

Power for the tags is a function of user specification based on ping frequency, and the resulting power will be affected. CC1310 has compatibility with coin battery operation that has predicted at 10 years with low energy modes. For sensors, The MSP432 has various operating modes that enable low power, and will be stimulated by a solar panel and voltage regulator. The MSP432 also has dual modes such as 80uA/MHz active and 660nA RTC standby operation. For both the tags and the sensors, the FCC regulates transmission power to be 36 dBm, or 4 watts maximum effective isotropic radiated power (EIRP).

4.2.3.3 Input Voltage Level

The CC1310 tag will receive a 3.6V DC input from a coin cell battery. The same voltage will be seen on the field sensors for each of the CC1310s.

4.2.3.4 Input Noise and Ripple

The Launchpad versions of theCC1310 will be able to handle voltage swings, and so these effects will not be considered unless moving to PCB implementation.

4.2.3.5 External Commands

The Ranch Hand shall document all external commands in the appropriate ICD.

Rationale: The ICD will capture all interface details from the low level electrical to the high level packet format.

4.2.3.6 Outputs

4.2.3.6.1 Data Output

Tags will transmit their pre-programmed ID information in a series of packets under 2KB along with battery power information. Sensors will transmit Tag ID information with the additional time stamp and protocol information

4.2.3.6.2 Diagnostic Output

The interface at the home base will provide a troubleshooting ability to see the status of each sensor and tag in the system.

Rationale: Provides the ability to manually control things for debugging and a way to view/download the node map with associated potential targets.

4.2.3.7 Connectors

The output will be an Ethernet cable and it will be connected to the internet through the gateway router.

4.2.3.8 Wiring

Wiring will be specified appropriately for the connections in the tags and sensors.

4.2.4 Environmental Requirements

The Ranch Hand system will be able to operate under and withstand environmental factors commonly found in grazing fields where livestock roam.

Rationale: The requirements listed below conform to the standard of the system being placed outside in a pasture to monitor livestock.

4.2.4.1 Land Geography

Most livestock ranching fields do not vary drastically in land elevation with flatland being most common for pastures and grazing fields. However, for certain terrain with rolling hills, the system must be able to still appropriately detect and triangulate the location of tagged livestock in the field.

4.2.4.2 Thermal

The system must be able to operate in the outside heat of a pasture. Temperatures in this environment can range from 0°F to 110°F.

4.2.4.3 Pressure (Altitude)

The average altitude of livestock grazing fields in the United States range from 1,700 ft. to 3,000 ft. Therefore, the system shall be able to operate in atmospheric pressure ranges commonly found at these range of altitudes

4.2.4.4 Rain

The system shall be able to withstand rainfall both light and heavy such that common storms in the outdoors do not damage the system.

4.2.4.5 Humidity

The average humidity levels of livestock grazing fields range from 0% - 100% humidity. Therefore, the system shall be able to operate under these humidity levels with little disruption to its intended use.

4.2.5 Failure Propagation

4.2.5.1 Failure Mitigation

The Ranch Hand system shall not propagate failures beyond the bounds and interface of the Ranch Hand system. Several features of the system are meant to mitigate and accommodate failures of system components, and alert the rancher that something has failed.

4.2.5.1.1 Tag Battery Information

The Ranch Hand system shall monitor and transmit to the sensor stations the life of the battery within each tag, allowing the rancher to take measures to ensure that tags never run out of power. In addition, the Ranch Hand system will alert the rancher if a particular battery within a particular tag falls below 10% of the maximal charge, notifying the rancher that action should be taken immediately.

Rationale: This is a requirements specified by our customer due to constraints of their system in which the Search and Rescue System is integrating.

4.2.5.1.2 Sensor Station Battery Information

The Ranch Hand system shall monitor the battery of the individual sensor stations and shall send an alert to the base station if the battery falls below 10% of maximal charge, indicating some failure within the power system and allowing the rancher to perform maintenance.

Rationale: The system shall be able to detect power system failures within the sensor stations and notify the rancher.

4.2.5.2 Failure Detection, Isolation and Recovery (FDIR)

4.2.5.2.1 Loss of Communication with Tag

The Ranch Hand system shall notify the rancher if a tag has not transmitted to a sensor station for at least five minutes and the Ranch Hand System shall be able to provide the previous whereabouts of the tag.

Rationale: This is a requirement of the system. The system tracks tags and should alert the rancher if the tags become lost or penetrate some predefined boundary. (See Concept of Operations.)

4.2.5.2.2 Loss of Communication with Sensor Station

The Ranch Hand system shall notify the rancher if one of the sensor stations does not transmit and receive data properly, i.e. the base station system has not received any form of data from the sensor, for at least 20 minutes.

Rationale: The mesh sensor network is designed to continuously receive data every minute. A 20 minute timer would be an abnormal timespan for the system not to receive data. This timespan will be a good choice in detecting any anomalies with the system.

4.2.5.2.3 Sensor Mesh Network Routing Protocol Adjustment

In the event of the complete failure of a sensor station, the mesh network routing protocol of the Ranch Hand system shall be able to adjust the data routing such that data can still be transmitted through the remaining components of the mesh network, albeit at some lower but still operational functionality.

Rationale: The routing protocol should be able to conform to the particular setup of sensors within the mesh network and therefore should be able to adjust for such a failure. (See Concept of Operations.)

4.2.5.2.4 Built in Test (BIT)

The Ranch Hand system shall have a built-in (BIT) test program that shall run daily and generate tests signals. The built-in (BIT) test program shall detect if there are failures in the system or if there are communications quality issues, defined as communications quality falling below 90% for each sensor station in relation to the other sensor stations within the sensor mesh network.

Rationale: This is a requirements specified by our customer due to constraints of their system in which the Search and Rescue System is integrating.

4.2.5.2.5 BIT False Alarm

The Ranch Hand's built-in test (BIT) program shall have a false alarm rate of less than 5%.

Rationale: False alarms within a test should be limited as possible, and 5% would sufficiently limit the number of false alarms.

4.2.5.2.6 BIT Failure Log

The Ranch Hand system's program shall keep a log recording any errors detected during the running of the built-in test (BIT) program, and should the rancher wish to see this, shall be able to produce this failure log to the rancher.

Rationale: This is meant to provide the rancher, or whoever may be diagnosing the problem, with a better idea of any problems associated with the system.

5 SUPPORT REQUIREMENTS

Details of support are discussed and any requirements for the customer are also discussed in the sections below.

5.1 Customer Requirements

5.1.1 Power Supply

The customer will need a power supply to house the router and connecting items of the Base station.

5.1.2 Internet Facing Port

The customer will need to provide one of the following options for an internet gateway:

- Coax cable with Modem
- Ethernet port
- Internet hotspot

The Ranch Hand
Charles Anderson
Randy Ardywibowo
Jeff Bartlett
Connor Furqueron

INTERFACE CONTROL DOCUMENT

REVISION – Final
04/13/ 2016

INTERFACE CONTROL DOCUMENT

1. OVERVIEW

This document will discuss the various interfaces that the Ranch Hand Cattle Monitoring System will have. These interfaces include but are not limited to physical, electrical, thermal, communications protocol and device interfaces. Additionally, all three component parts (the base station, tags, mesh network algorithm, and sensor stations) will have their own interfaces between the component parts and the environment, and these interfaces will be discussed separately. First, any references used in this document will be given and all terms used in this document will be defined. Then, the physical interface of each of the component parts will be discussed. Afterward, the thermal interface will be presented. After the thermal interface, the electrical interface will be defined. Finally, the communications protocols and device interfaces will be discussed.

2. REFERENCES AND DEFINITIONS

References used in this document and definitions for terms used in this document are given below before any interfaces are discussed

3. REFERENCES

MIL-STD-810F
Environmental Engineering Considerations and Laboratories Tests
1 Jan 2000
Change Notice 2
30 Aug 2002
American National Standard for VME64 (ANSI/VITA 1-1994 (R2002))
4 Apr 1995
American National Standard for VME64 Extensions (ANSI/VITA 1.1-1997)
Oct 1998 **Definitions**

CCA	Circuit Card Assembly
mA	Milliamp
mW	Milliwatt
MHz	Megahertz (1,000,000 Hz)
TBD	To Be Determined
GUI	Graphical User Interface
PC	Personal Computer
USB	Universal Serial Bus

4 PHYSICAL INTERFACE

5.2 Weight

5.2.1 Tags

The Ranch Hand's tag consists of the Simplelink™ CC1310 Launchpad, a 3.5V coin cell battery, and a case for protection. The total will weigh .6 lbs.

5.2.2 Field Stations

With the accompanying solar panel, the Ranch Hand's sensors will weigh 4 lbs.

5.2.3 Base Station

The home sensor will weigh the same as the sensors dictated in 3.1.2, but with an accompanying home PC which the client will provide.

5.3 Dimensions

5.3.1 Tags

The Ranch Hand's Simplelink™ CC1310 Launchpad is 3.75" x 2.30"x.07" (**Error! Reference source not found.**) . Combined with a coin battery and the protective cover, the total tag should be 4"x3"x.5".

5.3.2 Field Stations

The field station body will be fairly small with two CC1310s, and battery pack. This component without the accompanying solar panel will be 8"x5"x3". With the solar panel, this will be the added dimension of the solar panel.

5.3.3 Home Station

This will be the same as in 3.2.2, minus the solar panel.

5.4 Mounting Locations

5.4.1 Tags

Tags will be mounted to livestock, either at the top of the hip or the head.

5.4.2 Field Stations

Field Stations shall be mounted at a height 2 feet higher than the tallest livestock for optimal Line of Sight.

5.4.3 Home Station

The Home Station will be wherever the place of best access to an internet facing hardware.

6 THERMAL INTERFACE

6.1 Environmental Temperature

Since the tags and sensors will be placed outside in a pasture field, the tags and sensors must be able to operate under temperatures ranging from 0°F to 110°F. To accommodate for these temperatures, the subsystems will be passively cooled by air circulation. If, through testing, it is found that air circulation is not sufficient to cool the sensors and tags, then an alternative method of cooling will be integrated into the system.

6.2 Battery Heat

To control the heat around the battery pack on the sensor stations, insulation will be wrapped around the battery pack to prevent it from overheating. The battery packs will also be placed underground so that direct sunlight does not ruin the lithium ion battery pack.

7 ELECTRICAL INTERFACE

7.1 Primary Input Power

7.1.1 Tags

Primary input power for the tags shall be provided by a coin cell battery, supplying power to the CC1310 via the CC1310 Launchpad's onboard power electronics circuitry.

7.1.2 Field Stations

Primary input power for the sensor stations will be provided the two CC1310s. However, these shall share common power and ground as the two Launchpad units will be electrically connected together. Power shall be provided to the combined Launchpad modules through physical wire connection from the Texas Instruments BQ25570 Charge Controller Module. This shall control electrical power coming from the photovoltaic cells and the lithium ion battery pack. The flow of charge and power from the lithium ion battery pack and the photovoltaic cells will be mediated by the BQ25570 through physical wire connection.

7.1.3 Home Station

A normal electrical wall socket will be sufficient to power this system.

7.2 Polarity Reversal

7.2.1 Tags

Polarity Reversal protection in the tags should be done in the manufacturing process with correct assembly. In order to provide small and lightweight tags, minimal additional circuitry shall be provided with the CC1310 Launchpad and coin cell battery; thus, no polarity reversal protection shall be included.

7.2.2 Field Stations

The internal circuitry of the BQ25570 will prevent damage to the components of the sensor stations in the event of reversed polarity. If these onboard circuits prove insufficient, additional protective circuitry may need to be supplied in the sensor stations to protect the CC1310.

7.2.3 Home Station

Internal Circuitry will be sufficient.

7.3 Signal Interfaces

7.3.1 Tags

Signal interface between the sensor stations and the tag on the part of the tag will be facilitated by the onboard antenna of the tag's CC1310.

7.3.2 Field Stations

Signal interface between the sensor stations and the tag and the other member of the sensor mesh network shall be mediated by the onboard antenna of the sensor station's CC1310. Communication between the CC1310 Launchpads of the sensor station will be done through physical linkage of the two launchpads together, since the two Launchpads can be placed on top of each other and linked through the pins on the Launchpads

7.3.3 Home Station

Not only will there be a webpage for any locally connected computer to access to administer to the website, but there will be Ethernet access as well.

7.4 User Control Interface

The user control interface will be provided through a Graphical User Interface (GUI) at the client PC as part of the Client PC software package. This GUI will be displayed on the computer's screen and will take input from the computer's keyboard and mouse or mousepad. This user control interface will be open source and developed using Electron, a graphical user interface library freely available for open source projects. This library allows our application to be implemented to numerous different operating systems, as well as web browsers without additional development for porting.

The application will be programmed using a Model-View-Controller (MVC) design framework. The model for this framework will be the sensor data, while the views and controllers will be the user interface of the application.

7.4.1 Triangulation and Ranch Mapping

The user interface shall gather the distance data transmitted from the sensors and transform them into cattle locations in a ranch map interface.

7.4.2 Tag Battery Level Notification

As the data sent from the tags will include battery level data, the battery level of each tag can be displayed on the user interface. The user interface shall give an indicator of the battery life of each tag and notify the user whenever the battery life of a tag is lower than a certain threshold.

7.4.3 Heat Map of Herding Location

The location of cattle over time can be recorded and used to generate a heat map indicating their general congregation location.

7.4.4 Tag Inactivity Warnings

The user interface shall detect whether a tag has not moved in a certain amount of time. This inactivity can be attributed to possible abnormalities in the system. This includes tags that are detached from the cattle or possible injuries to the cattle preventing their movement.

7.4.5 Tag Out of Bounds Warnings

The program shall warn the user when a tag goes out of the bounds specified by the sensors. This bound is determined by the outward most sensors in the ranch perimeter.

8 COMMUNICATIONS/DEVICE INTERFACE PROTOCOLS

8.1 33-cm Band Broadcasting (902 to 928 Mhz)

Per the FCC, there are maximum output power requirements for an antenna or transmitter at these frequencies.

> *Several of the FCC part 15 rules govern the transmit power permited in the ISM bands. Here is a summary of those rules:*

1. Maximum transmitter output power, fed into the antenna, is 30 dBm (1 watt).

2. Maximum Effective Isotropic Radiated Power (EIRP) is 36 dBm (4 watt).

You can obtain the EIRP by simply adding the transmit output power, in dBm, to the antenna gain in dBi (if there is loss in the cable feeding the antenna you may subtract that loss).

3. If your equipment is used in a fixed point-to-point link, there are two exceptions to the maximum EIRP rule above:

§ *In the 5.8 GHz band the rule is less restrictive. The maximum EIRP allowed is 53 dBm (30 dBm plus 23 dBi of antenna gain).*

§ *In the 2.4 GHz band you can increase the antenna gain to get an EIRP above 36 dBm but for every 3dBi increase of antenna gain you must reduce the transmit power by 1 dBm. The table below shows the combinations of allowed transmit power / antenna gain and the resulting EIRP.*[1]

8.2 U.S. Federal Government Frequency Allocations

5.150 The following bands:[2]

13553-13567 kHz	(centre frequency 13560kHz),
26957-27283 kHz	(centre frequency 27120 kHz),
40.66-40.70 MHz	(centre frequency 40.68 MHz),
902-928 MHz	in Region 2 (centre frequency 915 MHz),
2400-2500 MHz	(centre frequency 2450 MHz),
5725-5875 MHz	(centre frequency 5800 MHz), and
24-24.25 GHz	(centre frequency 24.125 GHz)

are also designated for industrial, scientific and medical (ISM) applications. Radiocommunication services operating within these bands must accept harmful interference which may be caused by these applications. ISM equipment operating in these bands is subject to the provisions of No. 15.13.

US218 The band 902-928 MHz is available for Location and Monitoring Service (LMS) systems subject to not causing harmful interference to the operation of all Federal stations authorized in this band. These systems must tolerate interference from the operation of industrial, scientific, and medical (ISM) equipment and the operation of Federal stations authorized in this band.

US267 In the band 902-928 MHz, amateur stations shall transmit only in the sub-bands 902-902.4, 902.6-904.3, 904.7-925.3, 925.7-927.3, and 927.7-928 MHz within the States of Colorado and Wyoming, bounded by the area of latitudes 39° N and 42° N and longitudes 103° W and 108° W.

US275 The band 902-928 MHz is allocated on a secondary basis to the amateur service subject to not

[1] http://www.afar.net/tutorials/fcc-rules

[2] http://www.spectrumwiki.com/wiki/display.aspx?f=902000000&limit=on

causing harmful interference to the operations of Federal stations authorized in this band or to Location and Monitoring Service (LMS) systems. Stations in the amateur service must tolerate any interference from the operations of industrial, scientific, and medical (ISM) devices, LMS systems, and the operations of Federal stations authorized in this band. Further, the amateur service is prohibited in those portions of Texas and New Mexico bounded on the south by latitude 31° 41' North, on the east by longitude 104° 11' West, and on the north by latitude 34° 30' North, and on the west by longitude 107° 30' West; in addition, outside this area but within 150 miles of these boundaries of White Sands Missile Range the service is restricted to a maximum transmitter peak envelope power output of 50 watts.

The Ranch Hand
Charles Anderson
Randy Ardywibowo
Jeff Bartlett
Connor Furqueron

SUBSYSTEM DESCRIPTION DOCUMENTS

REVISION – Final
04/13/ 2016

Mesh Network Subsystem

By Jeff Bartlett

1 INTRODUCTION

The following document is a representation of the Mesh Network Subsystem implemented by the TI sponsored Livestock Tracking team. This subsystem is a result of the established Concept of Operations (ConOps), Functional System Requirements (FSR), and Interface Control Document (ICD) that were made at the beginning of the semester. This section will first establish what was accomplished from each of those documents. Each of the requirements will be proved in the subsystem analysis in Section 2.

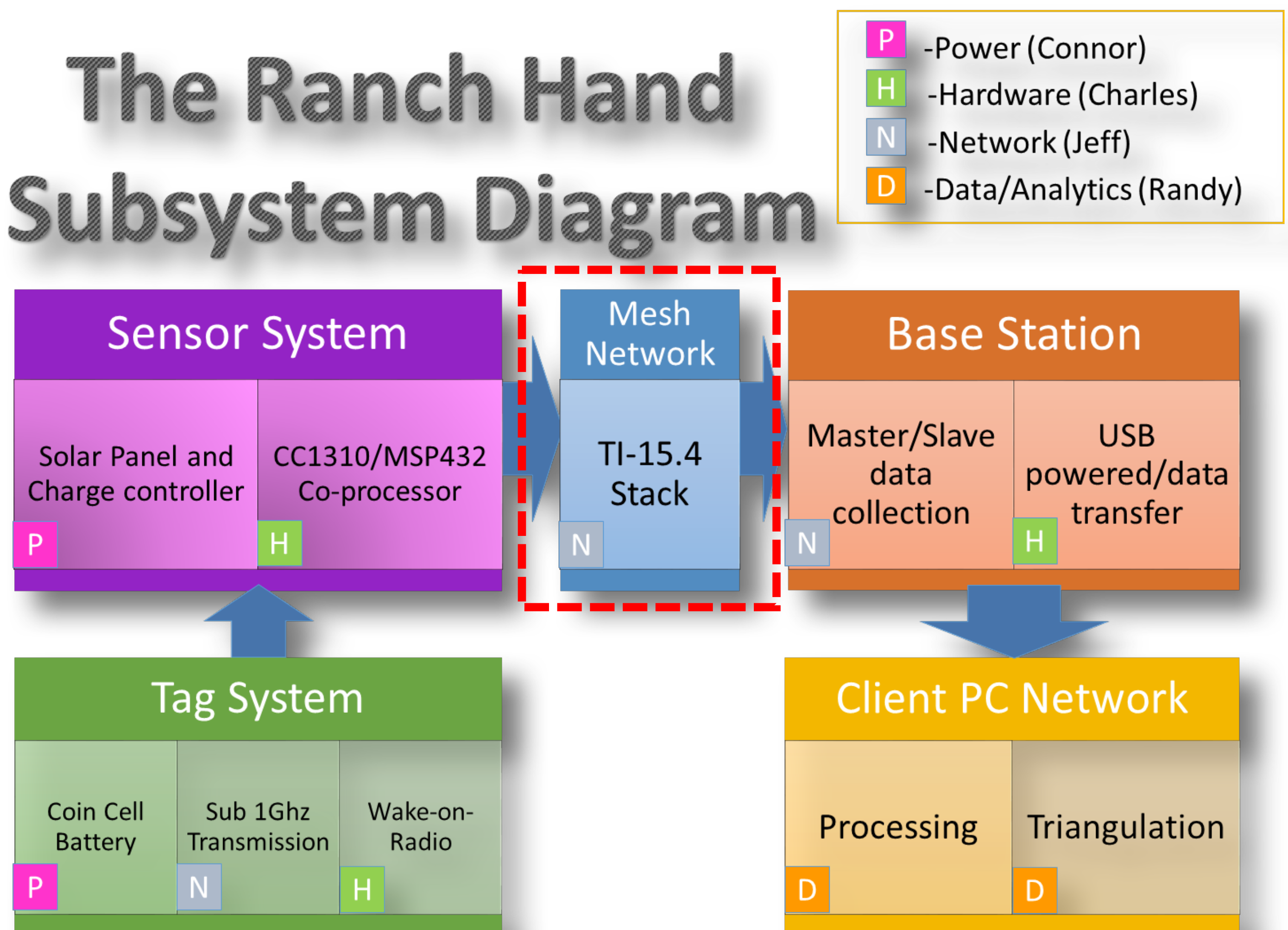


Figure 5: The Ranch Hand System

Concept of Operations

The concept of The Ranch Hand as far as the Mesh Network Subsystem (MNS for short) is concerned deals primarily with the concept of routing. The network described is one that will be able to cover an area appropriate for the user and use TI's parts. There was an established need for field sensors to pick up tag signals and a collector to which they would be sent. Specific requirements were further established in the FSR.

Functional System Requirements

The Functional System Requirements document further delineated what the mesh network needed to do in order to fulfill its role in the project. These are listed below for the reader's convenience:

a) Minimum Coverage Area- The system shall have a minimum area of coverage of three sensors of 300 square meters.
b) Terrain Contingency- The system will function optimally in terrain where the elevation does not vary more than plus or minus 5 meters and where there are more than 2-3 barriers per m^2.
c) System Capacity-The system should be able to track at least 24 tags simultaneously
d) Dimensions-Tags shall not weigh in excess of $1/4^{th}$ of a pound and not exceed physical dimensions of 2-½''x5''x1''. Sensor stations shall not have a weight in excess of 6 lbs and shall not exceed the physical dimensions of 14''x11''x5''.
e) Range- Tags shall be detectable by the sensor stations up to 100 m away. Sensor stations should have a transmission and reception range of up to 1 km.

These expectations were either met by the system or proven to be met by future planned operations. This will be explained in greater detail in Section 2.

Interface Control Document

The Interface Control Document gave requirements regarding the size of the system in order to make it compatible with the Power System. Signal interfaces were included in this and will be discussed in the following sections (see MAC packets).

Validation Plan

Below is the initiated plan that was followed for the semester.

Phase I: Hardware

a) Order MSP432, CC1310, and UART USB dongle

   The MSP432 and CC1310s are used for the sensors and the UART USB dongle is used to test the UART connection between the two launchpads by first sending it to a computer to read through COM port on the PC.
b) Test data transfer and flashing of MSP432, CC1310

   This will be important to make sure I have familiarity with CCS developer, the RTOS for the TI launchpads, and how they can be flashed and programmed.
c) Test UART dongle with MSP432 loaded program

   Use the USB dongle to transmit a message to the PC and read through the serial terminal.
d) Test CC1310 Launchpad range
    a. Range test

       Make sure to run multiple tests regarding the CC1310 and the PCB antenna to make sure the range is appropriate for the system and fits the FSR.

b. Battery test

Attach the CC1310 to a coin cell battery and see if the Launchpad can be powered in order to test remote capabilities.

c. Packet sending test

Test the CC1310s with the SmartRf Studio to make sure they can send designated packets and collect them appropriately.

Phase II: Mesh Network Setup

a) Research into 6lowpan and network implementation

6lowpan is a network stack that runs on mesh network nodes to help with network routing and implementation. Initial research on the website shows that TI hardware can be flashed and use these networks. This might need some development if we are using CC1310 rather than the CC2650s that some of the network examples used.

b) Research into contiki mesh network tester and developer

Contiki is a linux based mesh network OS that can be developed and flashed through Ubuntu. Contiki includes Wireshark Packet Sniffer and Cooja. Cooja is a network simulator that can simulate node traffic in a mesh network and routing for the 6lowpan network.

c) Try Wireshark Packet Sniffer

Wireshark Packet Sniffer is meant to monitor the mesh network to see packet transmission and reception. This is integrated in the Contiki program that will be important once the network is implemented.

Phase III: System Integration

a) Design/Integrate/Test Software for MSP432 and CC1310

Test software for the MSP432 and CC1310 in order to confirm that they can transmit information accurately in the Contiki defined network. This is a network example so Wireshark Packet Sniffer could be used for this example.

b) Set up/Test PC communication

Use PC to test network communication using the linux based edge router. The network should appear and the network should be confirmed as such.

2. FIRST SEMESTER ACCOMPLISHMENTS

CC1310 Testing and Development

Having the TI 15.4 network established, the CC1310s had to be evaluated via testing and validation in order to make sure this network simulation would work and fulfill the requirements. These experiments were conducted with Charles Anderson.

The first test was that of the range of the CC1310s. This was a rough test of the range to determine the space that the network would have to cover. Although the data is rough, the determined range was that 200m could be used as the maximum distance for the sensors. It was enough to determine that our Launchpad's PCB antennas would be enough to substantiate a mesh network of 300m^2. The information is reiterated in Figure 6 and Figure 7, which were collected from our field test. This information was sent at 625 bps in Long-Range Mode with a 54 byte payload message. There were a total of 100 packets sent, and the receiver was able to detect how many were sent and missed.

Testing Distance (ft)	Testing Distance (m)	Average RSSI (dBm)	Received OK (# of Packets)	Received not OK (# of	Packet Error Rate (%)	Bit Error Rate (%)
98.4252	**30**	-48.7	100	0	0.00%	0.00%
131.2336	**40**	-63.4	100	0	0.00%	0.00%
164.042	**50**	-59.2	100	0	0.00%	0.00%
196.8504	**60**	-65.1	99	1	1.00%	0.01%
229.6588	**70**	-65.5	99	1	1.00%	0.01%
262.4672	**80**	-68.9	99	1	1.00%	0.01%
295.2756	**90**	-67.4	100	0	0.00%	0.00%
328.084	**100**	-70.4	100	0	0.00%	0.00%
360.8924	**110**	-65.1	100	0	0.00%	0.00%
393.7008	**120**	-70.1	99	1	1.00%	0.01%
426.5092	**130**	-77	92	8	8.00%	0.07%
459.3176	**140**	-74.8	99	1	1.00%	0.01%
524.9344	**160**	-68.5	100	0	0.00%	0.00%
590.5512	**180**	-73.7	99	1	1.00%	0.01%
656.168	**200**	-80	94	6	6.00%	0.05%
787.4016	**240**	-81.8	100	0	0.00%	0.00%
918.6352	**280**	-88	78	22	22.00%	0.21%
984.252	**300**	-87.3	15	51	77.30%	1.23%

Figure 6: CC1310 Range Testing Data

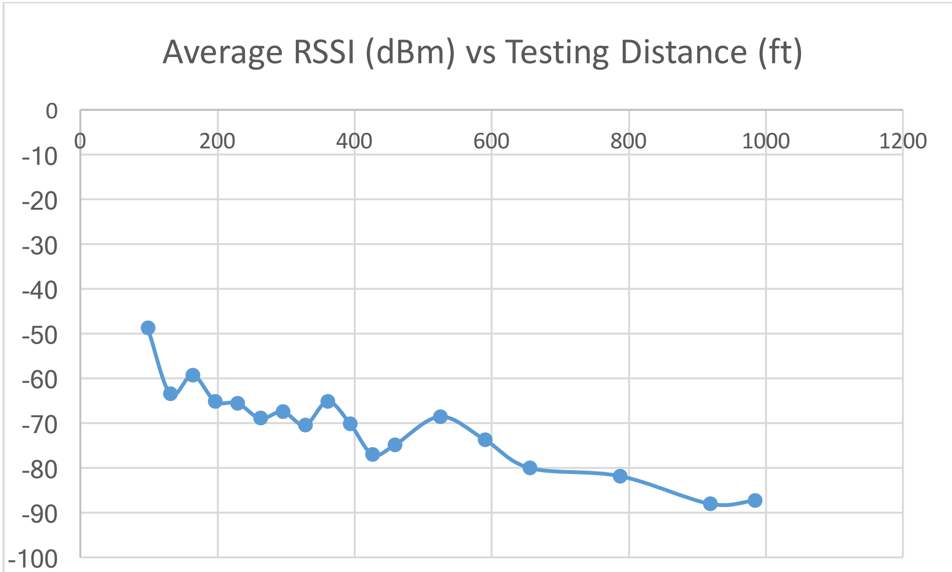


Figure 7: RSSI versus Range (ft)

This range test does not necessarily reflect the optimal range of our sensors, but rather a lower limit. This test was conducted in a parking lot and cars might have created interference. This suspicion was confirmed with the TI range testing model developed and shared on the sites E2E forum. An example showing a range capable of 800m is shown below. This was not realized by our test but still superseded our FSR expectations (Figure 8).

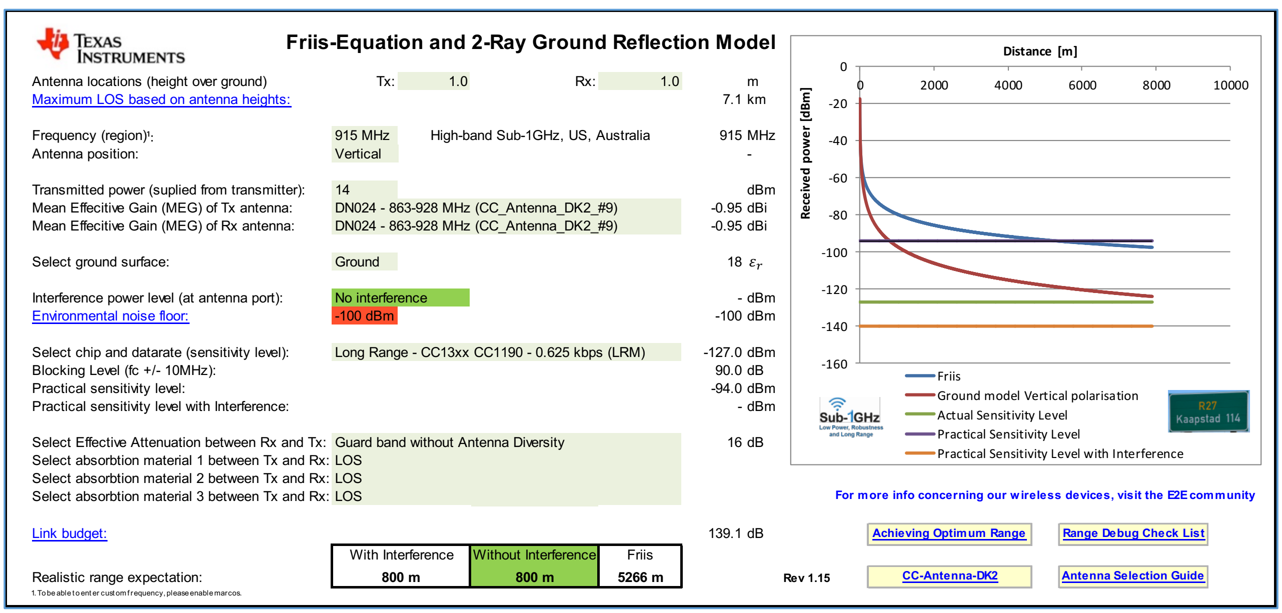


Figure 8: TI E2E Range Determination

Network Establishment

While developing the Contiki app as stated in the Validation plan, there were developmental problems with the network simulation app. The CC1310 was deemed not compatible with the network simulations. After this I moved to a different and proprietary network before the author informed me that the program similarly could not be ported due to library issues. After these setbacks, I was able to talk to the TI contact and work with the TI 15.4 star network stack. While not an official mesh network, the TI 15.4 can be used to send information from sensor nodes back to the collecting node. In the case of the Ranch Hand, this information would be tag information from the field nodes.

The Mesh Network Subsystem is based on the TI 15.4 Stack, a TI based star network that can be edited to fit user needs on wireless TI products. This network is the basis for the Ranch Hand's mesh network, which works with the tag system to turn the star network into a mesh.

The TI-15.4 Stack software architecture is based in IEEE 802.15.4, which is the technical standard for local wireless networks and can be extended in a Sub 1-Ghz range [1]. On the OSI model, the IEEE 802.15.4 network encompasses the physical (PHY) and data link layers (layers 1 and 2). The physical layer of this model is responsible for the RF transceiver and can make sure frequency hopping and channel selection is done properly. This is important to making sure the correct band is selected to quickly and effectively route the information back to base. The PHY layer also can edit modulation schemes and help corrective power procedures.

The data link layer is commonly referred to as the MAC layer, and enables the transmission of frames through the PHY layer channel. This is important to frame confirmation and arranging time stamping information. These layers are integrated in the TI 15.4 so that the application layer can be edited through the software development kit. The ICall layer in Figure 9 helps to interface the Stack and App in the software of the TI 15.4. ICall helps to organize different methods of communication via priority messages called semaphores.

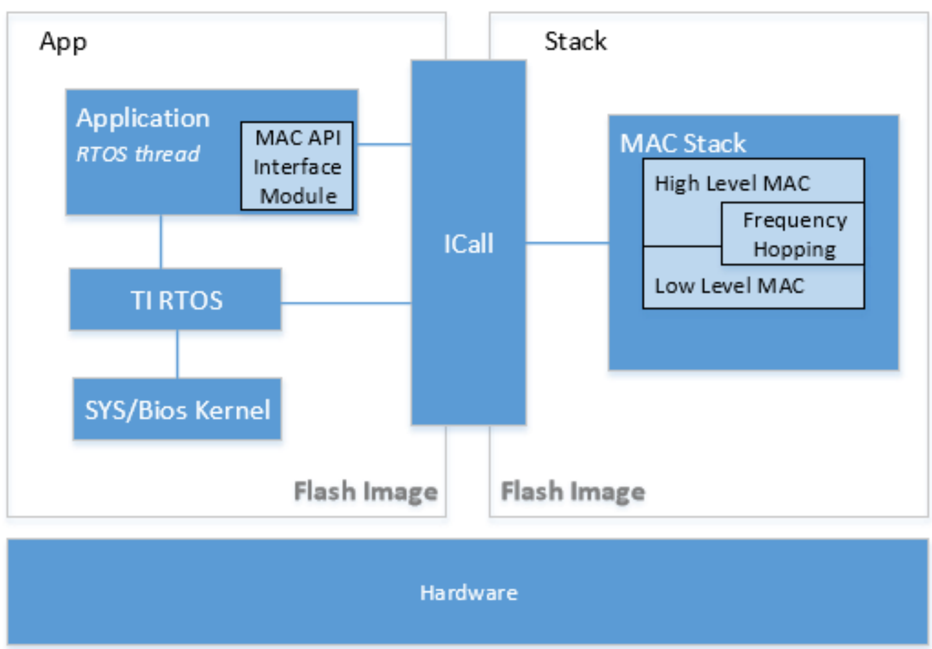


Figure 9: TI-15.4 Stack Architecture

Using this TI-15.4 program and the programs provided by TI, the network applications will follow either a sensor or collector application. The collector application dictates the node that will take all the data from the sensors. Likewise, the sensors will provide the data to the collector. Using these applications on the CC1310 launchpads I was able to send and collect MAC packets. There is also a method of integrating the mesh network sensors to sponsor a co-processor. What this means, is that the application and stack can be divided and flashed to different MCUs in order to split the processing. This is the function the MSP432 would play in the sensor as discussed in the ConOps, but the MSP432 is not a necessary part of this operation. If needed in further testing, the application can be ported to the MSP432 and the communication interface between the two will be UART based.

The next step in the network development was to link the collector to a PC interface. This was done using the Ti 15.4 Linux web guide, which can help to flash a UBUNTU running PC with a web application that uses a Node.js based script through which sensor information can be read. This example is pictured in Figure 10. The UBUNTU example provides access to the APIs to access the MAC layers and the application can be edited to provide tag information back to the collector.

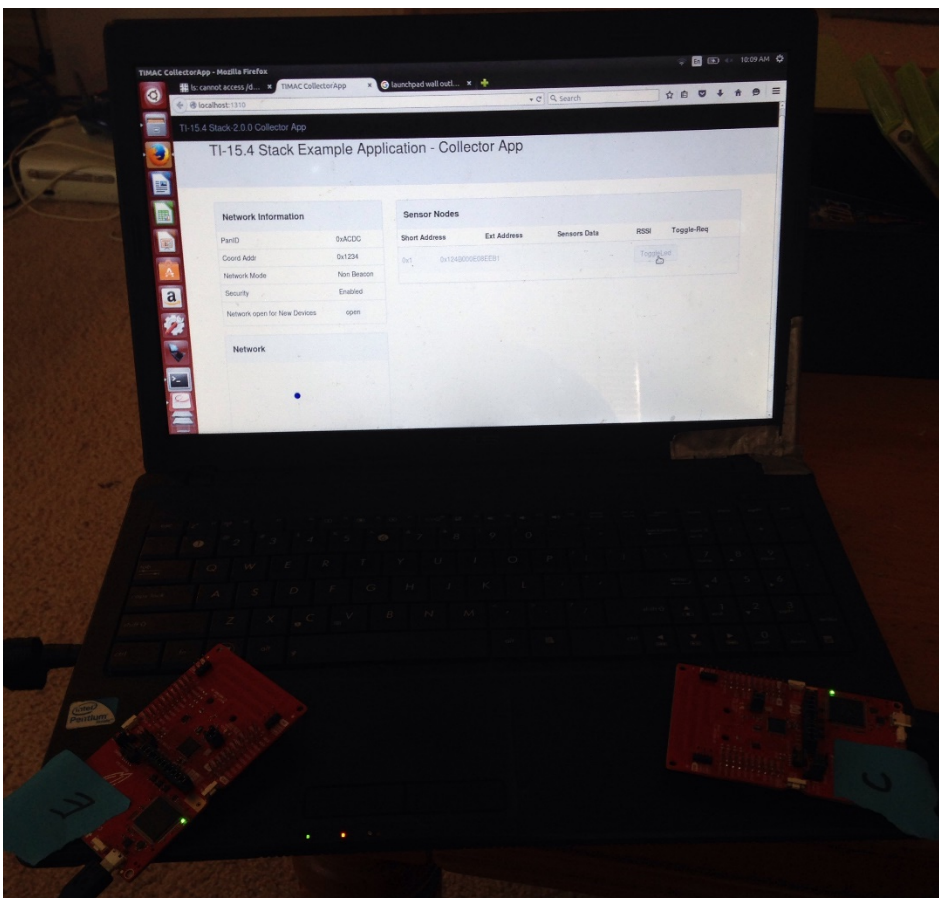


Figure 10: Example of Mesh Network and Linux Collector

Linux and TI 15.4 Development

The TI 15.4 stack uses two application functions that can be used by The Ranch Hand. One is a Toggle function, which in the example is used to blink an LED. This button is pushed through a Node.js application and the “sleepy” CC1310 wakes up to receive this information. The sensor application waits for the preamble and once detected and read will use the message. This helps to lower the power consumption of the sensor and will similarly be used on the Tag system.

This Toggle application is being developed to interface with the MAC API layer in order to beacon a specific tag/number of tags to have them wake and send beacon information. The information will be sensed and sent back in a listening slip-radio CC1310. A slip-radio method is where a separate radio is attached to the MCU in order to separately handle network information. A CC1310 slip-radio helps as a separate radio that can communicate over UART with the sensor. Charles Anderson has developed that system and the designs will be rectified in one unit next semester.

3. FSR/ICD REPORT

After the aforementioned analysis, I will now review the FSR requirements and confirm how they were fulfilled by the Subsystem.

a) Minimum Coverage Area- *The system shall have a minimum area of coverage of three sensors of 300 square meters.*

This requirement is more than satisfied. In fact, based on our 200m achievable distance, the computable coverage area is more than 28,000 m^2. This is because the coverage is laid for three sensors to cover what is known as a Reuleaux triangle. Randy Ardywibowo’s network simulations will greater explain this method.

b) Terrain Contingency- *The system will function optimally in terrain where the elevation does not vary more than plus or minus 5 meters and where there are more than 2-3 barriers per* m^2.

The subsystem has been confirmed to be small enough to fit on a pole that could be fixed for maximum line of sight distance. This means that no matter the terrain, the sensors can be moved for maximum coverage.

c) System Capacity-*The system should be able to track at least 24 tags simultaneously*

The TI 15.4 stack can handle more than 50 sensors by itself. By using CDMA or tag identification methods, the network can comfortably support enough tags to satisfy this requirement.

d) Dimensions-T*ags shall not weigh in excess of* $1/4^{th}$ *of a pound and not exceed physical dimensions of 2-½’’x5’’x1’’. Sensor stations shall not have a weight in excess of 6 lbs and shall not exceed the physical dimensions of 14’’x11’’x5’’.*

Sensor size was kept to the size of 2 launchpads and Connor Furqueron’s Power Subsystem. The size of the panel and system is actually smaller and lighter than predicted.

e) Range- *Tags shall be detectable by the sensor stations up to 100 m away. Sensor stations should have a transmission and reception range of up to 1 km.*

While the first requirement is met, the same spreadsheet used to help determine the Launchpad transmissions also helped us to realize that with a non-PCB antenna, distances of 1 km will be achievable.

4. SECOND SEMESTER ADDITIONS

During the second semester of the subsystem design, a complete overhaul was needed for the mesh network subsystem. There were several problems with using the TI-15.4 solution for the mesh system:

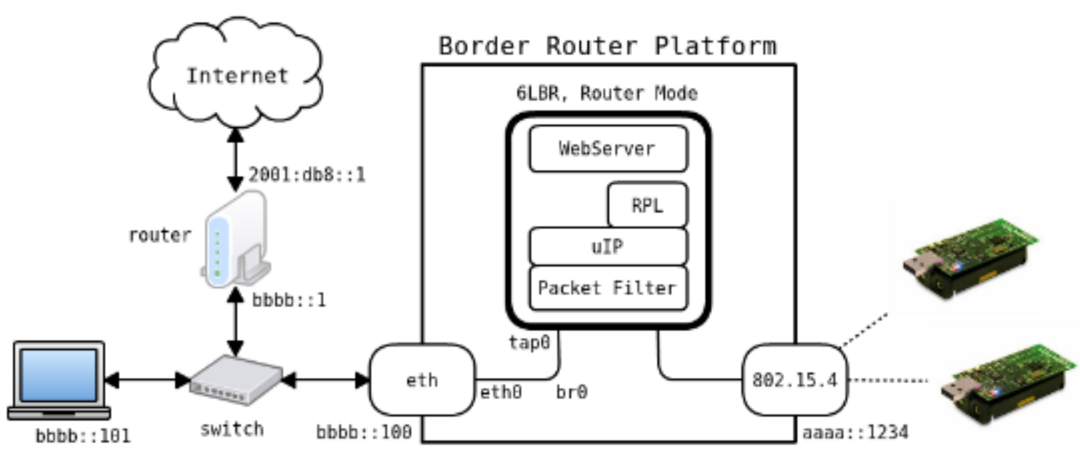
1. Mesh capability-TI-15.4 was a star network, meaning all data fed back to the collector, but there was no expansion capability that could be incorporated without major software overhaul.

2. Hardware compatibility- At the time of this writing, the CC1310 and MSP432 were not a compatible host for the software at the time of this writing. Therefore, there would have to be a set standard for porting the information over.

3. Information Routing- There was no implemented way of transferring sensor information to a web service that Randy could use. Ideally there could be an interface created through which the information could be routed so Randy could perform analytics.

The solution to this system was an OS called Contiki. This was a third party operating system that was compatible with TI hardware, gave full mesh capability, and was able to make every sensor IP addressable. The system is in short described below.

Router

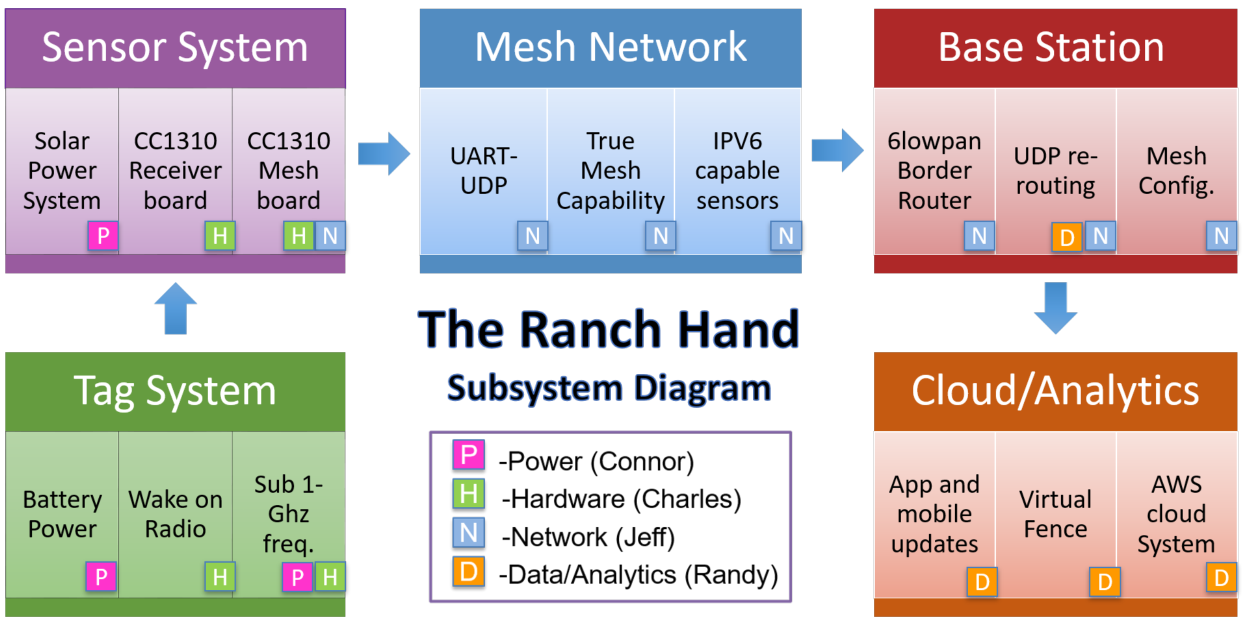


In the figure, there are mesh sensors that are picked up by a 6lbr (6lowpan Border Router) and sent over Ethernet to a switch/router. The beauty of the system as an IOT solution is that it can route sensor information over IPV6 to any address, and could be used with the AWS app that Randy had developed.

In addition to routing, the true mesh meant that the nodes could connect and route to the 6lbr router via other nodes. This "multi-hop" method allowed for future mesh expansion and therefore greater range. The mesh nodes were low-powered, with radio duty-cycles that would periodically check the airwaves rather and could save power.

The Contiki OS also allowed for future improvements of our system, such as an application called UART-UDP, in which UART information seen on the sensors peripheral could be routed via UDP over IPV6 to any address specified by the administrator. This would eventually be the method used to route RSSI information over to the cloud for analysis.

Because the Contiki system was now being implemented instead of TI 15.4, there was a need to change the entire system architecture. The update to the system looks like the following.



The procedure for this semester was divided into milestones. All of mine were met according to the table below.

Milestone	Responsibility	Description
M1: Subsystem Restart		
	Mesh Restart	Startup RPI and Confirm
M2: System Integration I		
	6lbr/sensor confirm	Access webpage with BBB
	Antenna characterization	Find power state relative to antennas for optimization
M3: System Integration II		
	Mesh to Cloud	Connect mesh sensor data to AWS
	Sensor to Mesh	Connect mesh sensor with receiving boards

	Antenna	Help to characterize and order antennas
M4: Small Model Testing		
	Mesh routing validation	make sure all data back to AWS
	Antenna	Antenna implement
M5: Model Expansion I		
	Mesh expansion	Sensor capacity and timing for sensor/tag comm
M6: Model Expansion II		
	Range test	See sensor/tag distance with power modes
	Prepare Field Test	Total system testing
M7: Final Testing		
	Debug and finalize	debug and finalize

In the M1 stage, the system needed to be transferred to a Beaglebone Black (BBB) with a CC1310 slip-radio. After flashing the software, I used an old router to connect the system via Ethernet and finish the border router. This took some time understanding the interfacing with the border router, and configuring the device's settings to what our system required. There was also a need to understand the UART interface and buying a USB dongle to override the driver problems faced with the Beaglebone.

After the 6lbr on the BBB was established, in M2 I had to attach sensors and be able to effectively talk to each of them through the 6lbr interface. After flashing all sensors, I had problems connecting to each sensor and the website on a local link, IPV6 web interface. After troubleshooting, I installed a NAT64 translator so that the IPV4 internet my ISP provided could talk to the Contiki IPV6 interfaces. This meant I could now see sensor information on the 6lbr administration interface.

6LBR
6Lowpan Border Router

System | Sensors | Status | Configuration | Statistics | Administration

Info

Info

Hostname : beaglebone
Version : 1.5.x (Contiki-contiki-base-develop-20170120-2242-g7b1fbac)
Mode : RPL ROUTER
Uptime : 0h 0m 22s

WSN

MAC: CSMA
RDC: br-rdc (0 Hz)
Security: nullsec
HW address : 0:12:4b:0:e:9:f:10
Address : fd00::212:4b00:e09:f10
Local address : fe80::212:4b00:e09:f10

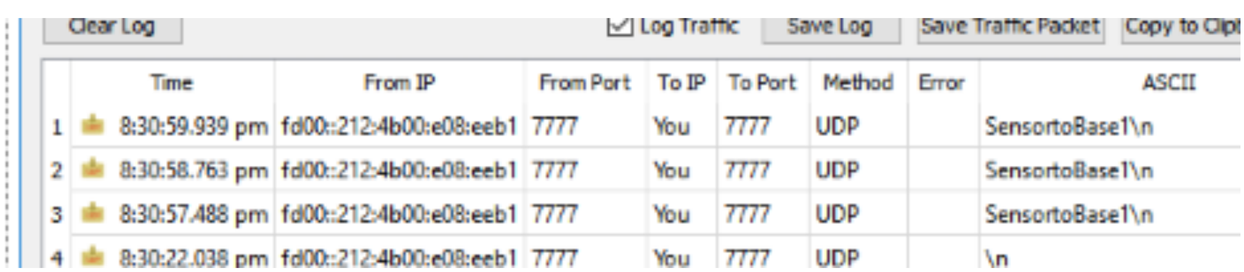
Ethernet

HW address : 6c:ec:eb:8a:fc:19
Address : bbbb::100
Local address : fe80::6eec:ebff:ff8a:fc19

6LBR By CETIC (documentation)
This page sent 1 times (0.19 sec)

During the M2 stage, I also assisted Connor in characterizing an alternative to the PCB antenna. The PCB antenna, while testing with adequate range, lacked the Omni-directionality needed to pick up accurate RSSI information in a 360 degree arc. Finding the impedance network and the antenna gain that seemed to match the system, Connor was able to order three sets of antennas that could work for us.

M3 was the first attempt at system integration with Randy's Amazon Web Service, and Charles's Tx/Rx tag system. The integration of both systems relied on the UART-UDP system, which I chose because of the IPV6 capability and the simplicity of UDP packet sending. After debugging with the handshaking interfaces between the TI Easylink UART and the Contiki UART, we found that using /n characters could signal the opening and closing of the UART. Through the Contiki admin page, I could specify the port number and IP address I wanted to send to.

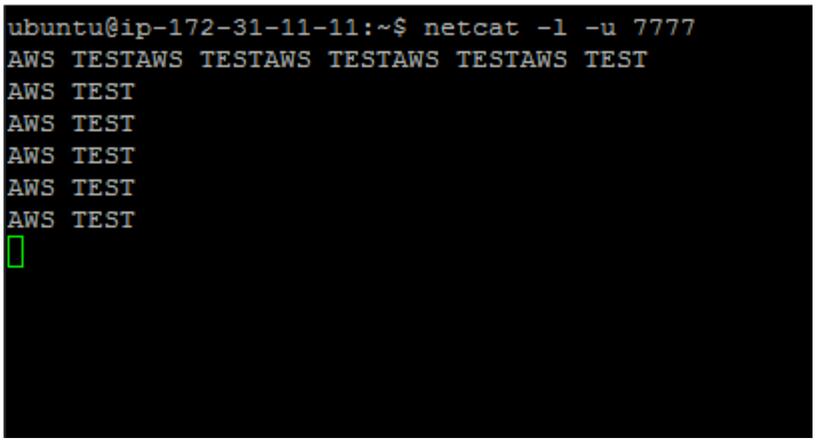


Clear Log | Log Traffic | Save Log | Save Traffic Packet | Copy to Clipb

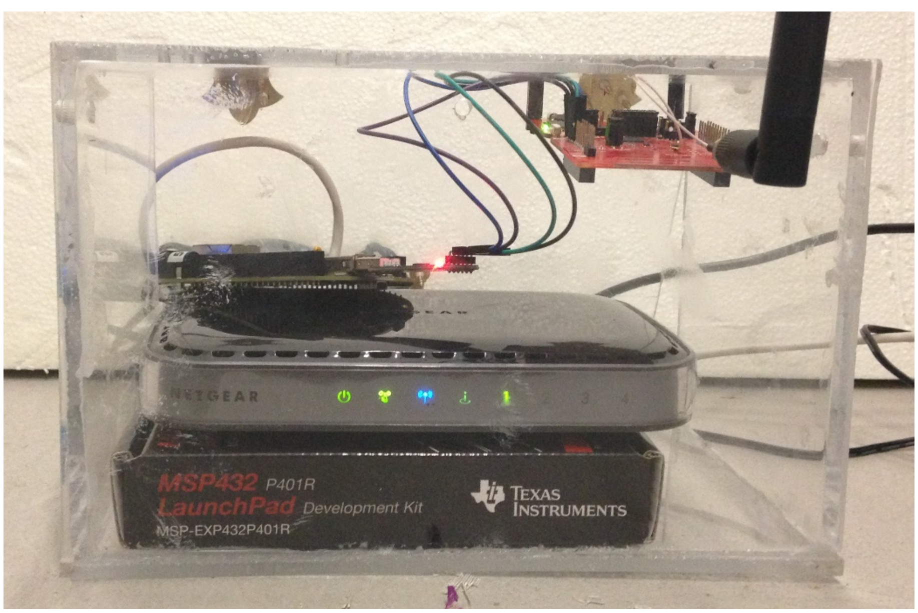
	Time	From IP	From Port	To IP	To Port	Method	Error	ASCII
1	8:30:59.939 pm	fd00::212:4b00:e08:eeb1	7777	You	7777	UDP		SensortoBase1\n
2	8:30:58.763 pm	fd00::212:4b00:e08:eeb1	7777	You	7777	UDP		SensortoBase1\n
3	8:30:57.488 pm	fd00::212:4b00:e08:eeb1	7777	You	7777	UDP		SensortoBase1\n
4	8:30:22.038 pm	fd00::212:4b00:e08:eeb1	7777	You	7777	UDP		\n

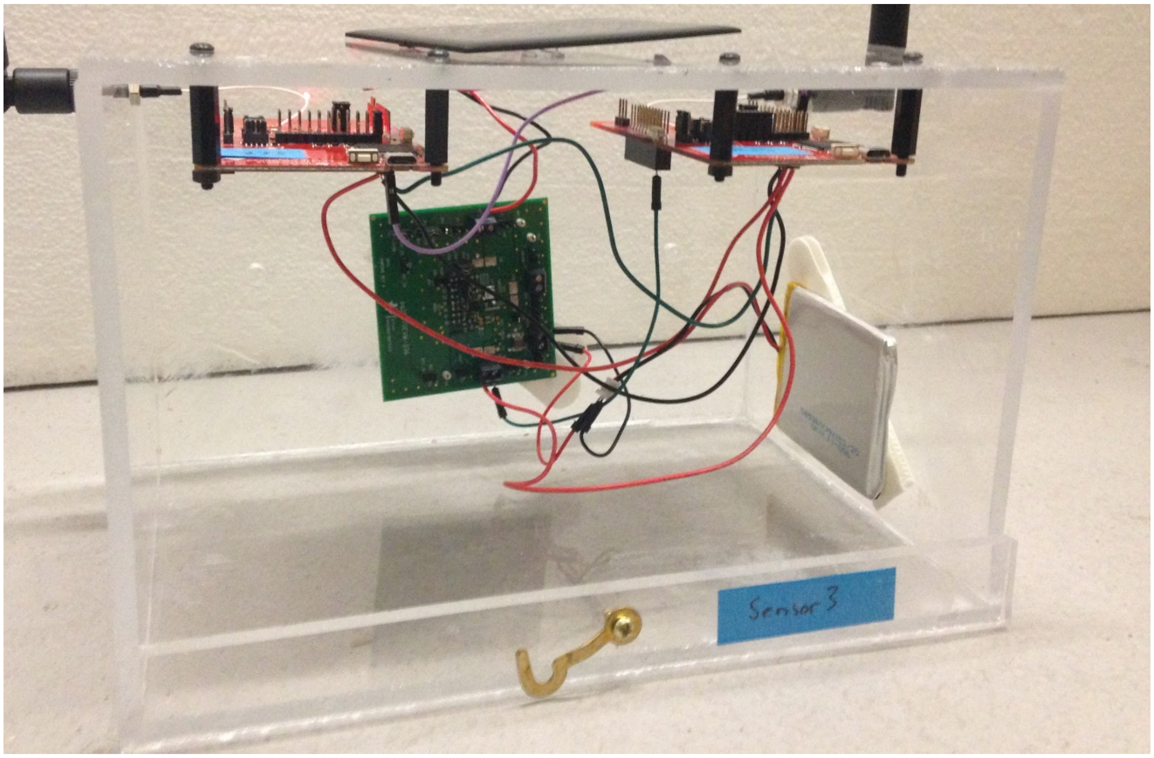
Using a serial dongle on the CC1310 sensors, I could see the sent packets on a UDP port listener on my computer. This confirmed that once Charles finished his code and used the proper characters discovered earlier, all we had to do was connect our boards via UART. Then, when a packet was received on one board, it would publish to UART, and the mesh board would receive it and send it via UDP to my computer.

The next integration was with Randy's AWS service. A major issue was how we were going to send IPV6 information over an IPV4 network to an IPV4 port. After reading into NAT64, I was able to translate the IPV4 address of the AWS to IPV6 and send the test packets from the sensors to a specified port on AWS. Listening with netcat port listener, I could see the following output:



M4 was a continued effort on expansion, making sure that Charles's code was compatible with the interface to make sure that RSSI information could be sent, and that on AWS we could tell the difference between the tags and sensors that they sent from. This was also the point when the antennas came in, so Connor and I were able to start using and testing them. We decided on the 6" duck antennas that had a 10dbm advantage over the smallest antennas. This was also the time when we used Connor's acrylic housing to build the housing systems. The base and the sensors fit easily into the enclosures that fit our FSR.





During the M5 stage, I began to test and validate systems with an expanded network. With the housing built, we began to characterize the mesh sensors and their range and strength within the network. We started to realize that we saw some strange readings when the mesh network was in a grassy area. It was then agreed that the noisier places to test were best because reflections gave the network redundancy and helped the communication between the sensors and the base.

	MESH CONNECTIVITY v. TEST LOCATION LOCATION							
	Track Complex		Arch. Quad (Grass)		EIC		Arch. Quad (Steps)	
Distance (ft)	RSSI (dB)	Index?	RSSI (dB)	Index?	RSSI (dB)	Index?	RSSI (dB)	Index?
3.28	-45	Yes	-47	Yes	-38	Yes	-41	Yes
6.56	-56	Yes	-55	Yes	-55	Yes	-52	Yes
16.4	-64	No	-62	Yes	-58	Yes	-54	Yes
22.96	LOSS	No	LOSS	No	-64	Yes	-54	Yes
32.8	LOSS	No	LOSS	No	-68	Yes	-57	Yes
49.2	LOSS	No	LOSS	No	-66	Yes	-68	Yes
65.6	LOSS	No	LOSS	No	-74	Yes	-68	Yes
82	LOSS	No	LOSS	No	-73	Yes	-72	Yes
98.4	LOSS	No	LOSS	No	-75	Yes	-74	Yes
114.8	LOSS	No	LOSS	No	-88	slow	-78	Yes
131.2	LOSS	No	LOSS	No	LOSS	No	-80	Yes
147.6	LOSS	No	LOSS	No	LOSS	No	-78	Yes
164	LOSS	No	LOSS	No	LOSS	No	-81	Yes
180.4	LOSS	No	LOSS	No	LOSS	No	LOSS	No

With this in mind, we tested our systems in terms of the new antennas and the different patches for the mesh systems. The list of tests are listed below that were performed during M6 and M7.

Category	Test	Description	Result
Tag	1.0-Tag Testing	Testing distance for the tag reception with respective antennas in busy area (@EIC)	Tested total packet reception up to approximately 400 feet
Tag	2.0-Tag Testing	Testing distance for the tag reception with respective antennas in open area (@track)	Tested total packet reception up to approximately 100 feet before packet loss
Field	1.0- Initial Field Test	Field Test 1: @track complex	web demo lossy, range loss on sensors, possible height sensitivity
Field	1.1-Secondary Field Test	Field Test 2: @ architecture Quad	web demo lossy, range loss on sensors, possible height sensitivity
Field	1.2-In House Test	Full test @EIC	simultaneous transmission from each sensor, all sensed at base without loss
Field	2.0-Field Test/Mesh Ranging	single mesh test with RDC patch @architecture Quad (steps)	Range up to 50m, depending on surroundings
Mesh	1.0-Mesh reflash RSSI testing	tested @EIC after initial reflash	Better RSSI
Mesh	2.0-Mesh Battery Test	tested @EIC after reflash 2.0	Better RSSI
Mesh	3.0-Three sensor mesh test	Ensure mesh capability with 3 boards	Confirmed mesh capability
Mesh	4.0-RF driver patching test	test null_rdc @EIC to test RSSI	Better RSSI
Mesh	5.0-null_rdc, and #1932 test	test null_rdc, pull request #1932, and @EIC to test RSSI	Better RSSI
Mesh	5.1-null_rdc and #1932, and VDDH_reg	test with null_rdc, pull request #1932, and max tx power @EIC	Better RSSI
Antenna	1.0-All Board Test	Test Rssi with all antennas @EIC	6" antenna has best RSSI
Antenna	1.1-Mesh Antenna Test	Test mesh sensor Rssi with existing build	Mesh RSSI is not ideal
Antenna	1.2-Tag Antenna Test	Test Tag RSSI to see distancing with antenna	Tag RSSI is acceptable

These tests extended from M5 to M6 and were essential to expanding the mesh in terms of range and stability. The conclusions from initial testing was that in an open field, the network was essentially too lossy to stabilize and get information at a reliable distance with omnidirectional antennas. There were 3 different patches used to improve these distances.

The first remedy was turning off the contiki rf duty cycle driver, therefore ensuring that more packets are received because the radio was "always on". This drains the battery, but it gave better perceive stability. The next fix was to use a different OS Contiki build that better suited the RF drivers of the TI boards. A third fix was to change a register value that regulated the Tx and Rx power of the boards. This meant the boards transmitted at 14dbm, higher than normal transmitting powers.

By the end of testing, the mesh network had full functionality, but lacked a distance depending on the location of the deployment. In the future, if this system is to be used, it is recommended that Contiki be better configured for these boards, or boards are used that better operate on longer distances in omni-directional patterns.

Power Subsystem

By Connor Furqueron

1. Introduction to Power Subsystem:

The Texas Instruments engineer, Mr. Goh, at the inception of the Ranch Hand project specified that some part of the project was to utilize solar power to perform energy harvesting. With this specification in place, there was a general drive to develop a photovoltaic power system to sustainably power the sensor stations for long periods of time. As a reference for the overall system construction, Figure 1 presents a simplified block diagram of all high-level components contained within the system. Additionally, Figure 2 shows a detailed diagram of the power system.

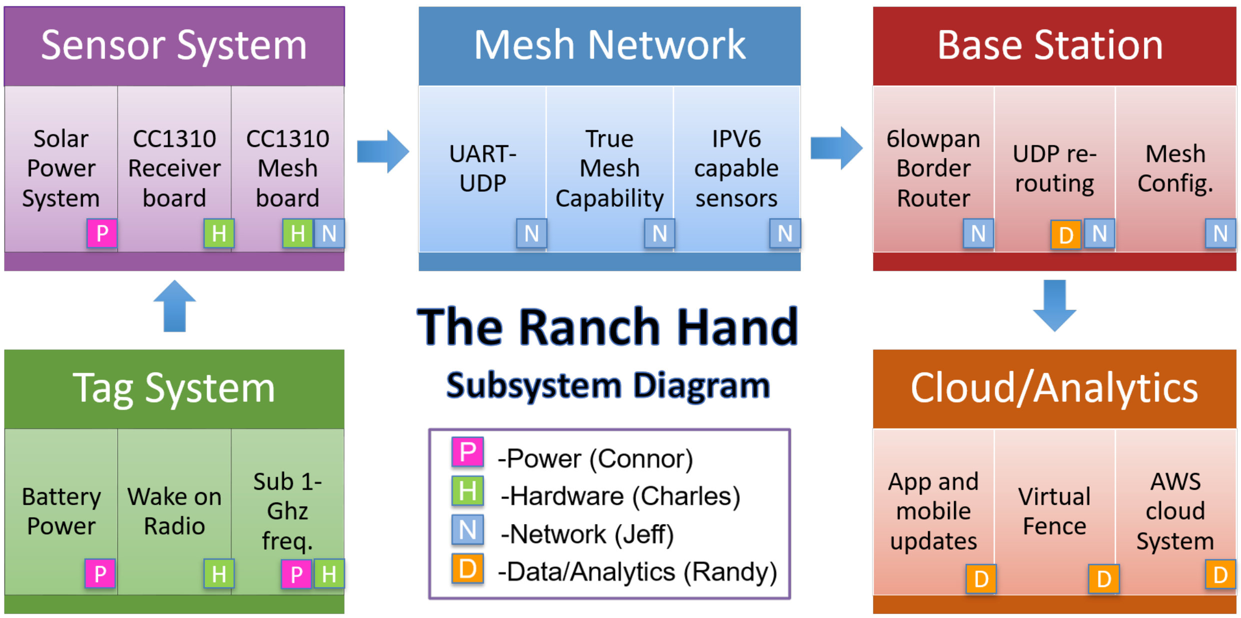


Figure 1: Overall Diagram for the Ranch Hand System.

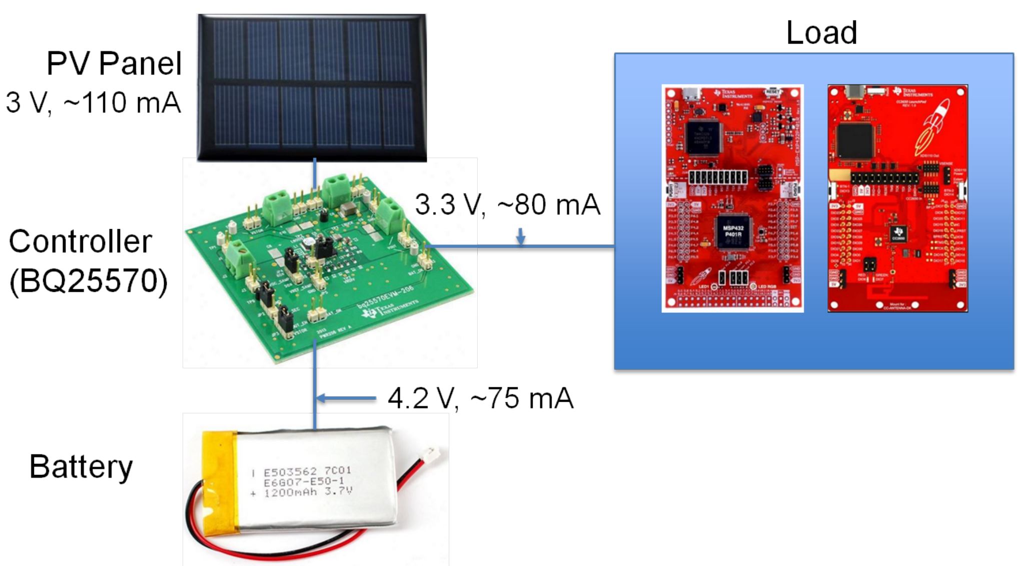


Figure 2: Detailed Diagram of the Power System.

As can be seen from Figures 1 and 2, the solar power subsystem powers the sensor nodes. The FSR laid out the requirements of the power system. The subsystem should have a theoretically unlimited life span based on power consumption and energy harvesting. Additionally, the subsystem should be able to power the CC1310s of the field sensors. Beyond this, the ICD laid out additional requirements on the power subsystem. A voltage of 3.1V to 3.7V must be able to be supplied to the CC1310s. This supply is accomplished through hard wiring the output of the charge controller to the power inputs of the CC1310s.

2. Construction of Subsystem:

The power subsystem is composed of three components: 1) a 2000 mAh, 3.7 V lithium ion battery, 2) a Texas Instruments BQ25570 power management charge controller, and 3) a 3V, 200 mA solar panel. The solar panel harvests ambient solar energy from the environment. This energy is then used to power the CC1310s of the sensor stations and charge the battery for nighttime system operation. The purpose of the battery is bipartite: first it fulfills the system's power needs when the solar panel cannot provide sufficient power, especially at night, and secondly, it accommodates short bursts of high current that the solar panel may not be able to supply to the CC1310s. The enabler of this scheme of power flow in the system is the BQ25570 power management charge controller. Its purpose is to mediate and manage the flow of power throughout the system. The primary focus of the BQ25570 is to supply the load, first with the photovoltaic panel and then with the battery. Any excess power generated from the photovoltaic panel is used to charge the battery. Overall, the BQ25570 made the job of designing the power system fairly simple.

Why were the particular components selected? For the BQ25570, the selection of this component actually came from the Texas Instruments engineer, Mr. Goh, who thought that this component was a very good solution for the power management of the system. Therefore, Mr. Goh sent several BQ25570 evaluation modules for the team to use. Since the selection of this component was accomplished by the Texas Instrument's engineer, the task of the team was to learn how to effectively use the BQ25570 for the purposes of the system.

In selecting the battery, different battery technologies had to be evaluated and the capacity of the battery had to be determined. In evaluation of battery technology, several considerations had to be made. The first and primary consideration was that the battery must be rechargeable. Secondly, it must be able to withstand multiple charging cycles. Finally, it must be fairly cheap to limit the price of the system. The CC1310 was created as a low power device, meaning that power consumption is low. Out of the multitude of available battery technologies, the lithium ion battery started to look like a particularly good option. The batteries, while more expensive than disposable batteries, were still fairly cheap for rechargeable batteries. Additionally, the technology was able to undergo a large number of charging cycles. However, two points of concern were the volatility and temperature sensitivity of the batteries. Fortunately, the BQ25570 has onboard facilities to help ensure battery safety. It provides undervoltage and overvoltage protection and monitors the temperature, safely controlling the flow of power into and out of the battery. With the concerns mitigated and the advantages of the technology observed, it was decided to utilize lithium ion technology. In estimating the size of the battery, initial estimates of the power consumption of the system over the course of a 15 hour nighttime period were about 500 mAh using pessimistic power consumption assumptions. Thus, to ensure that the system's power needs would be met for these 15 hour periods and to allow for the system to operate a longer than one nighttime period, a battery size of 2000 mAh was selected.

The selection of the solar panel was a much more convoluted endeavor that involved the making of a few mistakes. The first attempt at selecting a solar panel was done without truly understanding the purpose and input voltage range of the BQ25570 chip. Thus, a 6V, 5W solar panel which was too large was selected initially. This panel could not be used; thus, after going back and reevaluating the design goals, a 3V, 200 mA solar panel was selected. The pessimistic power consumption estimate put combined power consumption of the CC1310s at 0.2 W, while this panel could supply 0.6 W. Additionally, the 3V output voltage of the panel was well within the input voltage range of the BQ25570. The BQ25570 will draw a maximum current of 110 mA from the input source, so a panel with the potential to supply 200 mA would supply the BQ25570 with as much current as the charge controller could use. Thus, a 3V, 200 mA photovoltaic cell turned out to be a good choice. As an unintended positive consequence, while the sheer power generation capacity of the solar panel was sacrificed, the new solar panel proved to be sufficient while also being cheaper, lowering the price point of the power subsystem.

To understand the interaction of the component devices, Figure 2 shows the power subsystem with voltages and currents flowing between the components. As can be seen in the figure, the BQ25570 will draw out a maximum of 110 mA at 3V from the 3V solar panel. In reality, due to source resistance, this current will be closer to 90 to 100 mA. The BQ25570 will then supply a well-regulated charging voltage to the battery port of 4.2 V, and charge the battery with up to 75 mA, based on the load's current consumption and the source current output. To the load, a well-regulated 3.3V output can be supplied with up to 80 mA based on how much current is needed by the load. In instances where the load power consumption is less than the power being consumed by the solar panel, surplus power will be used to charge the battery. In situations where the solar panel cannot sufficiently supply the load, the battery will be used to supply the additional power.

Figure 3 shows the schematic of the BQ25570 evaluation module, allowing one to get a sense of the physical construction of the subsystem. The photovoltaic panel is placed at the input voltage terminal, Vin. The battery is placed at the battery terminal, Vbat. Finally, the load is placed at the output voltage terminal, Vout. This is how the subsystem is physically constructed. With the subsystem physically constructed, a multitude of data can be collected about its operational characteristics. The EVM was modified with R10 being changed to a 2.43 Mohm resistor, regulating the output voltage at 3.3V instead of 1.8V.

The tags placed on the cows were chosen to be powered by two AAA batteries, placed in series. This setup would supply the tags with 3V, which is sufficient to power the CC1310. Also, using AAA batteries will ensure that the tags are easily serviced and that the replacement batteries remain cheap.

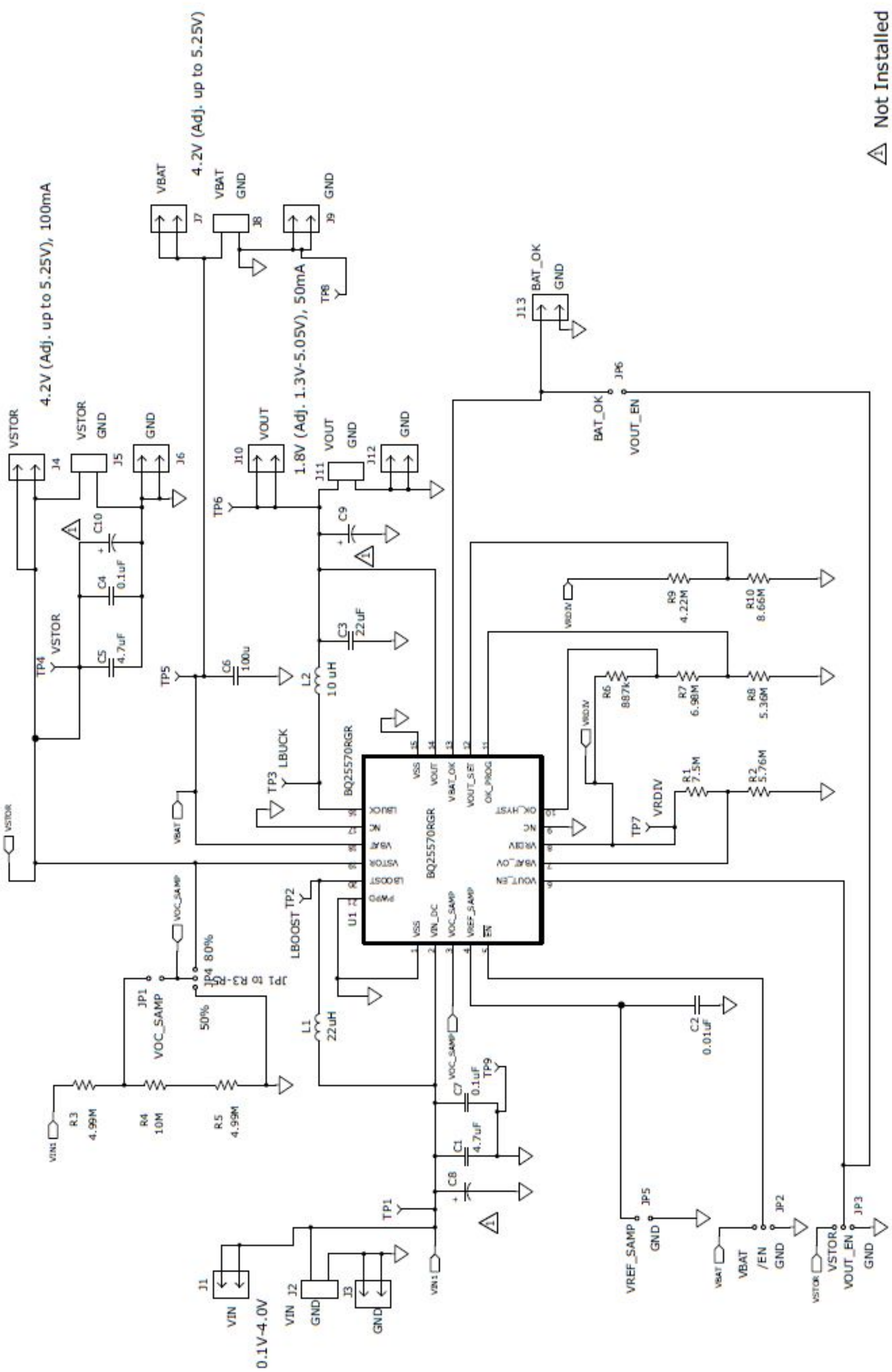


Figure 3: BQ25570 Datasheet Schematic.

3. Data Collected for Validation:

Data was gathered on the various components to help in validating the operational conditions of the subsystem. First a test of the behavior of the solar panel was done to understand how voltage and current were affected under a resistive load. The panel was loaded with 100 ohms and exposed to the afternoon sun at about 3 pm in the late part of October. The data collected is shown in Table 1. The angle of incidence of the solar panel to the sun was changed and the effect on the voltage and current was observed.

Solar Panel with Resistive Load Data (100 ohm)		
Angle	Voltage	Current
0deg	3.41V	0.033A
30deg	3.33V	0.032A
60deg	3.24V	0.031A
90deg	2.59V	0.025A
180deg	1.85V	0.016A

Table 1: Voltage and Current with Resistive Load

As can be seen from Table 1, both the voltage and the current started to drop significantly after 90 degrees, leading to an operational angel of about 120 degrees in which the sun can traverse and the panel can generate sufficient power. An interesting effect was also noted. If the panel was oriented at 180 degrees, and a large, light colored object was placed approximately 2.5 inches from the face, the voltage went up to 2.44V and the current increased to 0.022A. The object was essentially acting as a partial reflector, and it was reflecting more solar radiation onto the panel than the ambient environment. Although useful for understanding the behavior of the solar panel, the data collected in Table 1 does not really provide information on how the BQ25570 extracts power from the solar panel.

To understand how the BQ25570 extracted power from the solar panel, another test was performed later in the semester where the solar panel and the battery were emplaced at the Vin and Vbat ports of the BQ25570 respectively. The same angles of incidence were tested and the voltage and current at the Vin port were measured. The data collected results in Table 2.

Solar Panel Data with BQ25570		
Angle	Voltage	Current
0deg	3.00V	95mA
30deg	2.97V	80mA
60deg	2.94V	50mA
90deg	2.00V	10 mA
180deg	1.8V	0mA

Table 2: Solar Panel Test with BQ25570

Comparing Table 2 with Table 1, it is interesting to note that the BQ25570 did a better job of maintaining the voltage of the solar panel. The current was the most affected quantity during the test. A small change in the angle of incidence would affect the current, while the voltage would remain fairly constant. Significant voltage drop did not occur until after 90 degrees. However, current would suffer a fairly significant drop after 60 degrees. At 60 degrees, the current was almost halved. Despite this, the angle of operation seemed to remain at around 120 degrees. Thus, the sun should be able to provide sufficient power to the panel if the angle of incidence is less than 60 degrees. The difference in Table 2 is attributable to the fact that the BQ25570 is a maximum power point tracker, extracting the maximum power from the solar panel based on complex onboard circuitry.

Another small test was performed measuring the output voltage of the BQ25570. Given a range of resistances, and even some capacitances, the output voltage remained unchanged at about 3.3V, with a variation in 20mV. Finally, a CC1310 was connected to the BQ25570 and the output voltage dropped down to 3.29V. This proved a well-regulated output voltage, which is imperative for the functioning of the launchpads.

Finally, a battery of tests was run in a controlled environment, using a bench DC power source in place of the solar panel. A CC1310 with the emulation hardware was connected to produce a load that drew 75 mA. Current and voltage were measured at all three ports to gain an understanding of the behavior of the system. The data collected resulted in Table 3, shown below.

Power Subsystem Current and Voltage Data						
Test Case	Vin	Iin	Vbat	Ibat	Vout	Iout
CC1310 Powered, Source On	3.0V	110mA	4.2V	(-6mA) C	3.29V	75mA
CC1310 Off, Source On	3.0V	110mA	4.2V	(-59mA) C	3.31V	0mA
CC1310 Powered, Source Off	3.0V	0mA	3.97V	67mA	3.29V	75mA
CC1310 Off, Source Off	3.0V	0mA	3.97V	1.33mA	3.31V	0mA

Table 3: Voltage and Current on Several Test Cases

Several noteworthy things are seen in the table. Firstly, the power is shared between the load and the battery. While the load takes priority, any excess power generated is delivered to the battery. The battery is held at a constant charging voltage of 4.2V when not being used to supply the load. Second, the maximum power draw of the BQ25570 is 110 mA. It was seen in Table 2, that the maximum power draw of the solar panel was about 95 mA. This is due to the solar panel being a high impedance source. Next, the battery was only ever charged with a maximum of 67 mA. Thus, in combination with the maximum power draw of 95 mA from the solar panel, the maximum current put into the battery with the solar panel as the source would be around 60 mA. Finally, with no load and no source input, the battery supplied 1.33 mA to the chip to allow the BQ25570 to perform its duties.

In addition to performing tests, several estimates of power consumption and generation were done. The calculation of power consumption was done in terms of battery capacity, or milliamp-hours. To calculate, first the average current was calculated and then multiplied by a time. The average current was calculated assuming a short pulse of high activity and a long duration of very minimal activity that would form an operational period. Generally, a five minute period was assumed with a 30 second pulse of high activity. From this method, a power consumption of 169 mAh is to be expected over the course of a 15 hour nighttime period. During the daytime period, the solar panel, if assumed to be able to charge the battery with 60 mA for a

9 hour daytime period, should be able to produce about 540 mAh for the battery. Thus, it would take almost 20 hours to fully charge the battery from a minimal charge state. However, it is seen that the power generated during the daytime period far exceeds the energy consumed over the nighttime period. Additionally, both estimates for the energy generated and the energy consumed were designed to be pessimistic in nature. Optimally, the subsystem's overall performance would outstrip these estimates. Additionally, with a 1200 mAh battery, it is estimated that the battery would last longer than five days with no recharging from the solar panel. Furthermore, other optimizations of the system's power consumptions could reduce the nighttime energy consumption, extrapolating the life of the battery.

4. Validation:

Below, one can see the original validation plan put together to test the system and ensure that it was working as intended in the FSR and ICD. For Semester I, all of Phase I of the validation plan was completed successfully. The voltage of the solar panel was tested extensively and found to be within the range that was required within the validation plan.

For Phase II validation, the charge controller was found to have a very well-regulated output of 4.2 V for charging the battery, meeting the first requirement. Also, for the third requirement, the charge controller was found to have a well-regulated output voltage at almost exactly 3.3V, well within the range required by the launchpads. However, the second requirement for Phase II validation was not met. The battery was discovered to have an overdesigned capacity. While not being able to be charged to full charge from minimal charge within a day, the upshot of the battery's overdesigned capacity is that the system is estimated to be able to operate for up to several dozen days with no solar source input. While not meeting one of the requirements initially laid out in the validation plan, this effect was judged to be desirable and the estimation of a power surplus was judged a sufficient sustainability criterion for long-term operation.

Phase III validation deals with systems integration and overall system validation, so some requirements were not able to be addressed. However, the system was able to power both an MSP432 and CC1310 and an extensive study of their power consumption was made. Although the power consumption is a little more than was anticipated, the power consumption was still found to be less than the pessimistic power estimates done at the very beginning of designing the power subsystem.

Thus, one validation requirement of Phase II was not met because a design judgment was made. It was judged that having a very large battery capacity was much more advantageous than being able to completely charge the battery within one nine hour daytime period. In place of this requirement, it was judged that having the solar panel able to produce an energy surplus that could overcome the deficit generated during the night would suffice. It was estimated that the MSP432 and CC1310 would consume 169 mAh over 15 hours, and that the solar panel should be able to produce approximately 500 mAh. Thus, the new validation requirement was met.

In the second semester, the group's time was devoted to system integration. The power system was further validated and could effectively power the system for prolonged periods of time. There were not actual tests to ascertain how the power system performed, since the group deemed such information as secondary to the primary concern of hardware development. Thus, the power system, as it stood, was deployed to power the system, and it did so effectively.

Overall, with most of the requirements of the validation plan, the power subsystem has been significantly developed and is estimated to be more than sufficient to supply the needs of the sensor system. However, if the sensor stations are reengineered to force the range of the mesh network to be higher, the system may need to be modified to supply increased power to the CC1310 or whatever replacement is used.

Power System Validation Plan

Connor Furqueron

Semester I

Phase I:

- Test solar panel for consistency of voltage at various inclinations to the sun and different times of day to gain a complete set of data on the solar panel, and thus better understand its specific characteristics.
- With a load across it, such as a 1 Mohm resistor, the panel must be able to supply at least 2 V and not exceed 5.4 V, as well as verify that the unloaded solar panel does indeed have an open-circuit voltage of about 4 V. (Observed using data points.)

Phase II:

- With battery and solar panel connected through charge controller, ensure that charge controller is outputting at least 4.2 V while not exceeding 5 V to the battery.
- With battery at minimal charge, ensure that solar panel is capable of charging battery to full capacity (by measuring the 4.2 V maximum voltage) during one day time period, at non-optimal angle and for shortest time of day during the year.
- Ensure that the charge controller has regulated load output voltage within 3.1 to 3.3 V, a range of voltages that should work for both the MSP432 and CC1310 based on datasheets. Ensure this range using a variety of resistive and non-resistive loads (1kohm, 1Mohm, 100 microfarad capacitor in parallel with 1 Mohm resistor, and perhaps some other loads)

Phase III:

- Power the MSP432 and CC1310 up with the power system. Take measurements of current consumption and calculate power consumption. Compare with estimated power consumption from designing the system earlier in the semester. Verify that system supplies 3.3 V to MSP432 and the CC1310. (If this power is exceeded, look into options for bolstering power of system or decreasing power consumption of the MSP432 and CC1310.)

- Once power system has proven sufficient for powering the MSP432 and CC1310, run 24 hour test to prove that system can run maintenance-free for at least a day. Verify that battery is still charged up when system is checked the following day.
- Once daily test has been successful, run a prolonged test to ensure that the battery is not dying over the course of a week of operation.

Semester II

M1: Subsystem Restart

- Get system put back together and validated.
- Order additional parts for three whole power system and three tags.

M2: System Integration I

- Begin PCB design.
- Figure out power states for the CC1310 and MSP432
- Aide in finding antennas that power system can handle.

M3: System Integration II

- Finalize PCB design.
- Finalize antenna orders.

M4: Small Model Testing

- Aide group in systems integration tasks.
- Emplace antennas onto CC1310s.
- Begin power testing with finalized hardware.
- Receive PCB and test its functionality.

M5: Model Expansion I

- Aide group in systems integration tasks.
- Once PCB design is validated, have 3 more PCBs produced.
- Continue power system testing to ensure longevity.
- Get MSP432 operating as system executor.

M6: Model Expansion II

- Aide group in systems integration tasks.
- Build housing for sensor stations.

M7: Final Testing

- Aide group with final testing.

5. Expansions

Future expansions on the power system include the inclusion of the MSP432 as a system executor, which would monitor the health of the system. In this capacity, the MSP432 can check on the CC1310s and monitor the battery as a redundant watchdog to the BQ25570's battery voltage monitor. If the MSP432 detects issues with the system, it can take steps to preserve the system in the event of communications failure, device failure, or impeding power failure, and try to inform the mesh network of the issue. Finally, optimizing the functioning of the CC1310s and MSP432s through power states will yield benefits for power consumption. For the tags, the CC1310s must stay in Standby mode, waking up only when a signal is received. On the sensor stations, the two CC1310s must stay in Idle mode and the MSP432 must stay in Low Power Mode 4.5 until conditions exist for them to wake up.

One big concern with the power system for both the tags and the sensor stations is the additional power consumption from switching to the omni-directional antennas. Trying to extend the range of the tags and sensor stations while transmitting in all directions requires a significant amount of power. The initial power system was not designed to supply that much power, but sacrificing some of the range allowed the power system to be sustainable while accomplishing the majority share of the goals for the project. That is the sensor stations were a satisfactory proof of concept despite the reduced range. However, if it is decided to redesign the sensor station to increase the range, the power system may need to be redesigned as well, using a larger solar panel, a larger battery, and a charge controller sized for these larger components.

Another expansion for both the sensor stations and the tags is to custom design the hardware, forgoing the launchpads for a printed circuit board. This would significantly reduce the cost and the size of the packages for both the sensor stations and the tags. Some thought was given to doing this during the semester; however, the BQ25570 was difficult for the facilities at A&M to deal with and producing the printed circuit board at a private manufacturer proved prohibitively expensive. Thus, the group decided to use the CC1310 launchpads and the BQ25570 EVMs that were supplied by Texas Instruments to prototype the system.

With the design of the printed circuit board in mind, a design of the power system was attempted in Eagle circuit design software. Some errors still exist in the design, but they should be easily fixed; its design was abandoned to allow more time to work on producing the prototype system from the EVMs and launchpads. This design is shown in Figure 4. The design stayed close to the EVM with the exception of two changes. R10 is a 2.43 megaohm resistor instead of the EVM's 8.66 megaohm resistor, regulating the voltage at 3.3V instead of 1.8V. Also, a Zener diode was added to the input port, Vin, to add a certain amount of overvoltage protection.

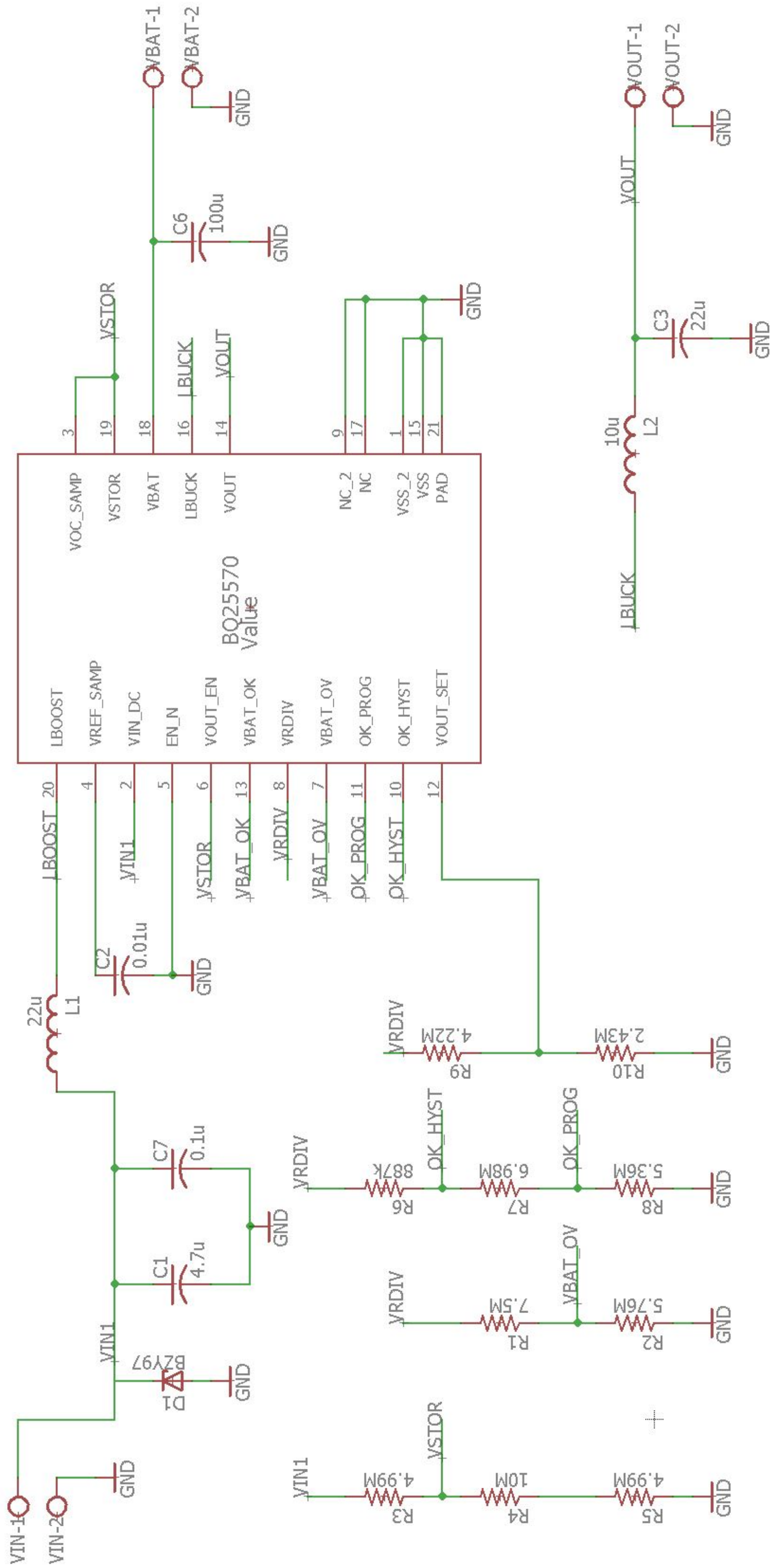


Figure 4: Power System Schematic Design.

Hardware Subsystem
By Charles Anderson

1. INTRODUCTION

This document describes the Tag and Sensor subsystem implemented by the TI sponsored Livestock Tracking team. The design of the subsystem is made in accordance with the specifications outlined in the Concept of Operations, Functional System Requirements, and Interface Control Document. The first section of this document recaps the expectations of the Tag and Sensor subsystem that were set by the three aforementioned documents. The second section of this document goes into detail about how the subsystem requirements were tested and validated.

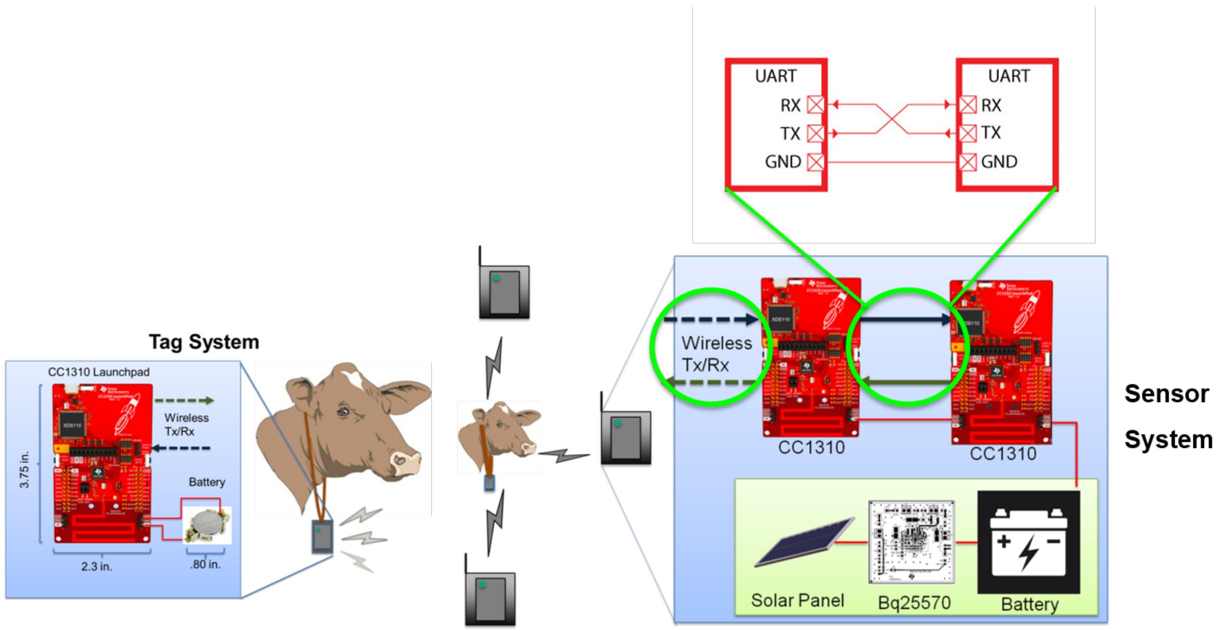


Figure 1: Hardware Subsystem Diagram

1.1 Concept of Operations

The objective of the Tag and Sensor subsystem as is described in the Concept of Operations document is to implement an active radio-frequency identification (RFID) system with a tag system acting as a transmitter (TX) and a sensor system acting as a transmitter and receiver (TX/RX). The tag system is powered by a coin cell battery and will transmit packets of identification information to a sensor station that will receive the packets, process the information received, then relay the information through the mesh network subsystem.

1.2 Functional System Requirements

The Functional System Requirements document describes in further detail the specifications the Tag and Sensor subsystem needs to satisfy. The tag and sensor requirements are listed separately below for convenience.

Tag component

- The tag system shall have a battery life of no less than 2 years.
- Each tag will be preprogrammed with a chip ID via USB on the CC1310 Launchpad.

- The CC1310 on the tags will receive a 900 Mhz signal from the sensors.
- The CC1310 tag will receive a 3.6 V DC input from a coin cell battery.
- Tags will transmit their pre-programmed ID information in a series of packets under 2KB along with an estimation of the tag's current battery power.

Sensor component

- The sensor system shall draw enough power from the lithium-ion battery such that it does not exceed the recharging capabilities of the solar panel attached to the sensor station.
- Sensors will receive a 900 Mhz signal from either a tag or another sensor.
- Sensors will transmit Tag ID information with the additional time stamp and protocol information.
- Sensor stations will notify the base station when a tag has not transmitted to a sensor station for at least five minutes.
- Sensor stations will be able to timestamp the reception of a transmitted tag signal to aid in triangulation calculation.

Both Tag and Sensor

- For both the tags and the sensors, the FCC regulates transmission power to be 36 dBm, or 4 watts maximum effective isotropic radiated power (EIRP).

1.3 Interface Control Document

The Interface Control Document specified that the Tag and Sensor subsystem uses a Serial Peripheral Interface (SPI) communication protocol in regards to the communication between the tag component and the sensor component of the subsystem. Further details on how SPI protocols were used in the subsystem are described in Task 2.

2 SUBSYSTEM OVERVIEW

The Tag and Sensor subsystem focused on implementing effective communication between a tag component, made up solely of a CC1310 microcontroller unit (MCU), and a sensor station, made up of a CC1310 MCU and an MSP432 MCU hooked up as a co-processor. The details of how the sensor station is linked up as a co-processor is described in further detail in the Mesh Network subsystem document.

The tag component of the subsystem is designed to transmit information in a Sub 1-Ghz range in accordance with the transmitting capabilities of a CC1310 MCU. The reasoning behind choosing a CC1310 MCU for transmission and reception of information is due to the TI sponsorship of this project specifically requesting that the CC1310 MCU be used in the project.

The tag component will transmit necessary information to the sensor stations such that when the information is relayed to the base station through a mesh network protocol, triangulation calculations can be done at the base station, a user's laptop or computer, that determine the location of the tag that transmitted the data relative to the sensor stations that received the initial signal. The necessary information to be transmitted was determined to be:

1) An identification number for the tag
2) An estimation of the battery level on the tag
3) The power level in dBm that the original signal was transmitted at

An identification number for the tag enables the base station to recognize tags uniquely and display the tag's information accurately on a user interface display. The estimation of the battery level on the tag enables the base station to store the information and display an estimate of a tag's current battery level to a user. The power level in dBm that the original signal was transmitted at is used to estimate the distance that the tag is from the sensor station that received the transmitted pulse of information from the tag. The drop in the power level of the signal when it is received by the sensor station can be measured by comparing the received power level to the transmitted power level. This drop in power level can then be used in computations that estimate the distance that the tag is from the sensor station. This distance measurement will then be relayed back to the base station through the mesh network subsystem for triangulation computations.

The sensor component of the subsystem is designed to receive and transmit information in a Sub 1-Ghz range in accordance with the transmitting capabilities of a CC1310 MCU. The sensor component is needed to receive information from the tag component and to transmit information across a mesh network protocol.

The sensor component will also operate in wake-on radio mode, whereby it waits in standby mode to conserve power until it detects a valid transmitted signal by reading the preamble of the transmitted signal. The sensor component will timestamp the time it receives a pulse of information from a tag and relay this timestamp information back to the base station to assist in computations.

2.1 Tag Component Development

The following sections describe how the various features of the tag component were developed and the tests that were performed to evaluate the performance of these features relative to what was needed of the tag in the Functional System Requirements document.

2.1.1 Transmit Packets at Sub 1-GHz

The first feature of the tag that is implemented is transmitting packets of information at Sub 1-GHz using the onboard antenna of the CC1310 Launchpad. Transmission is done using the TI-RTOS: Real-Time Operating System (RTOS) package for microcontroller units that is provided by Texas Instruments. TI-RTOS eliminates the need to create basic system software functions from scratch and enables development of features involving real-time on the CC1310 Launchpad. The TI-RTOS package provided libraries with functions that could be called upon to perform basic tasks on the CC1310 such as opening up a radio channel on the CC1310 with a specified frequency and power of transmission, transmitting packets of information across the radio channel, or receiving packets of information across the radio channel. Code Composer Studio (CCS) is used to edit and flash a program written using the TI-RTOS package to a CC1310 MCU. Two programs are written that can be flashed to two CC1310 Launchpads to simulate transmission and reception of packets of random information at a frequency of Sub 1-GHz.

2.1.2 Identification Number on Tag

An identification number for a tag is able to be transmitted across the Sub 1-GHz signal created by the program made using the TI-RTOS package. The program developed for the task of transmitting packets of information is modified to transmit packets of identification numbers instead of random information.

2.1.3 Low-Power Mode of Tags

The TI-RTOS package provides a function that can be utilized to put the CC1310 MCU into sleep mode, whereby it draws much less power than it normally would. This low-power mode enables the tags to emit a quick pulse of information data and then transition into sleep mode until the next timed pulse. A test was done to examine how much power was conserved when switching the CC1310 between active and sleep mode. The test involved programming a CC1310 to emit a pulse of information data over 1 second for every five seconds. During the other four seconds, the CC1310 would be placed in sleep mode. The results of this test are shown in Table 1.

Low-Power Mode Test on Tags			
During Transmission	Time of Transmission	Low-Power Mode over 4 seconds	Average Current for 5 seconds
0.815 mA	1 second	0.409 mA	0.4902 mA
0.810 mA	1 second	0.410 mA	0.4900 mA
0.809 mA	1 second	0.407 mA	0.4874 mA
0.812 mA	1 second	0.411 mA	0.4912 mA
0.810 mA	1 second	0.410 mA	0.4900 mA
0.811 mA	1 second	0.407 mA	0.4878 mA
0.809 mA	1 second	0.409 mA	0.4890 mA

Table 1: Results of Low-Power Mode Testing on Tag Protocol

From this test, it is concluded that the average current drawn by the CC1310 tag component over 5 seconds is 0.4894 mA.

2.1.4 Relay Battery Life of Tag

The battery life of the tag component is calculated from power estimations from the power subsystem. Given measurements of the current being drawn by the CC1310 Launchpad when it is transmitting packets and when it is not transmitting packets, an estimation on the remaining battery life left in the coin cell battery being used to power the tag can be calculated. Table 1 shows measurements on the average amount of current being drawn by the CC1310 Launchpad during transmission of data and when in low-power mode.

Given the capacity of energy in the coin-cell battery, an estimation is made on the lifespan of the battery with the average current drawn by the tag over 5 seconds. The lifespan of the battery is estimated to be 1 year with the average current of 0.4894 mA being drawn by the tag component over 5 seconds. With the lifespan of the battery estimated, the rate at which the percentage of the battery's life would deplete could be formulated. These calculations are integrated into the transmission program developed using TI-RTOS and CCS so that the battery level percentage calculation could be transmitted in the pulse signal along with the tag's identification number. Figure 1 shows the results of a sensor station receiving the packets transmitted by the updated program.

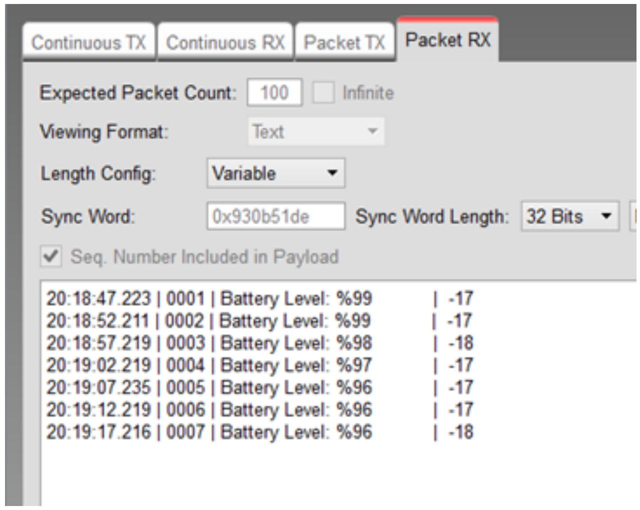


Figure 2: Display Showing Battery Level Received by Sensor Station

2.1.5 Relay Power Level of Transmission Signal in dBm

The original power level of the transmitted signal in dBm is determined to be a constant value. This is due to the value being solely dependent on the power level chosen to transmit the signal at. Thus, the value of the power level of the transmission signal is able to be integrated into the program in the same way the identification number of the tag is able to be transmitted by the program. This power level value will then be used by the base station to estimate the distance a tag is from a sensor station depending on the drop in power level between the received signal power and the transmitted signal power.

2.2 Sensor Component Development

The following sections describe how the various features of the sensor component were developed and the tests that were performed to evaluate the performance of these features relative to what was needed of the sensor in the Functional System Requirements document.

2.2.1 Wake-on Radio Mode

A wake-on radio protocol is developed to conserve energy on the sensor stations. The wake-on protocol ensures that sensor stations are active to receive a signal only when they detect a valid pulse signal sent by a tag and are on stand-by mode in the meantime. The protocol is based on the principle of duty-cycling the radio and entering RX just as much as is necessary to detect a packet in transmission. The protocol runs an RX Sniff command to check for the presence of a signal on the air with a receiver. The RX Sniff command checks the Received Signal Strength Indicator (RSSI) first. If the RSSI is not found to be over a set threshold in the RX Sniff command, then the receiver is set to go back to stand-by mode. However, if the RSSI is found to be above a set threshold, then the receiver checks for the presence of a valid preamble in the transmitted signal and notes the result of the check as the Preamble Quality (PQT).

The reasoning behind having the RX Sniff command check RSSI first before PQT is due to RSSI being quicker to check by the receiver. However, the RSSI gives less information as the reading only indicates that a signal is present and gives no qualitative information about the signal. Thus, by checking RSSI first, it can be determined whether or not a possible transmitted signal is in the air. Once a transmitted signal is found that is above the set RSSI threshold, a check on the PQT of the signal is done by the receiver. Preamble quality checks are done after RSSI checks due to preamble quality checks taking longer to check. Once a valid preamble is found in the transmitted signal, the receiver begins to receive the whole transmitted signal.

Tests were done to evaluate the energy conservation performance that incorporating wake-on radio mode into the sensor station program would have. The results of this testing are listed in Table 2.

Wake-On Radio Power Test on Sensor Station			
During Reception	Time of Reception	Standby Mode over 4 seconds	Average Current for 5 seconds
0.712 mA	1 second	0.112 mA	0.272 mA
0.690 mA	1 second	0.110 mA	0.266 mA
0.710 mA	1 second	0.109 mA	0.2692 mA
0.699 mA	1 second	0.111 mA	0.267 mA
0.695 mA	1 second	0.110 mA	0.267 mA
0.711 mA	1 second	0.109 mA	0.2694 mA
0.710 mA	1 second	0.105 mA	0.266 mA

Table 2: Results of Wake-On Radio Power Consumption Tests for Sensor Station

From the test results, it is concluded that the average current drawn by the sensor station while operating under the wake-on radio protocol is 0.268 mA. This current consumption over five seconds is quite reasonable for the application as the lithium-ion battery chosen to support the sensor stations is able to provide for this amount of current consumption over five seconds. More importantly, the solar panel will be able to compensate for this current consumption and still charge the lithium-ion battery during the daytime.

2.2.2 Timestamping

The TI-RTOS package includes various functions that allow the CC1310 MCU to handle real time computations and analysis. One such library is for timestamping that can be linked and called to allow the CC1310 to timestamp a time relative to a set time in Unix time format. Unix time format is the number of seconds between a particular date and the Unix Epoch on January 1^{st}, 1970 at UTC.

It was discussed whether or not timestamping is appropriate for this application due to timestamping producing a result with a unit precision in seconds. It was concluded that even though timestamping is unable to provide a result with unit precision smaller than seconds, the rate at which sensor stations will be receiving signals is approximately once every five seconds for the current design. Thus, trying to obtain a unit precision smaller than seconds was deemed unnecessary for the current design of the application. Figure 1 demonstrates a display of the timestamped values for every packet received.

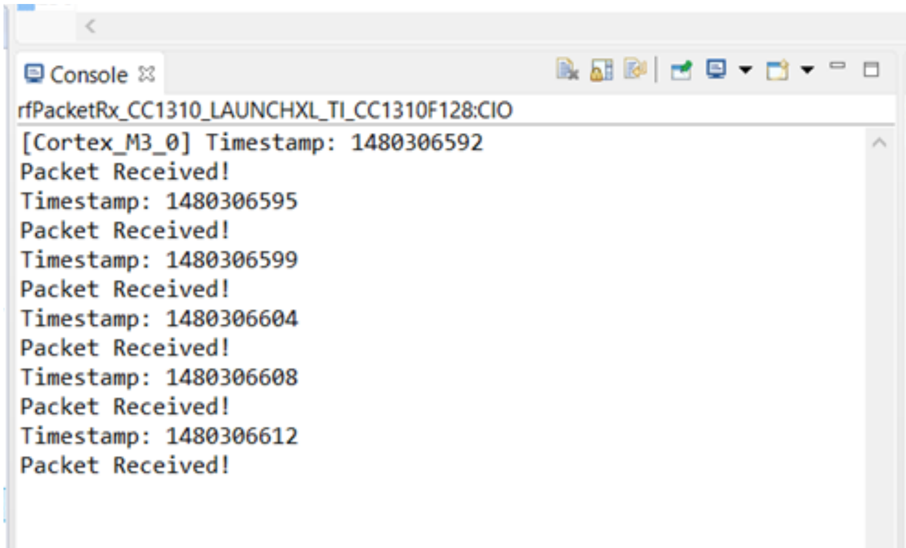


Figure 3: Console Display of Timestamp Values for Packets Received

The usage of the timestamp function in the TI-RTOS package also raised the question of how much time the timestamp function takes to run and if this delay in time would affect the timestamp resulting value. A test was done to investigate the time delay due to calling the timestamp function during operation of the CC1310 MCU. Table 2 is a collection of the results of these tests.

Test Number	Timestamp Delay
1	4.02 us (microseconds)
2	4.10 us (microseconds)
3	4.08 us (microseconds)
4	4.07 us (microseconds)
5	4.03 us (microseconds)
6	4.05 us (microseconds)
7	4.04 us (microseconds)
8	4.10 us (microseconds)
9	4.15 us (microseconds)
10	4.09 us (microseconds)

Table 3: Timestamp Delay of Calling Timestamp Function in Program

The average timestamp delay is computed to be approximately 4.073 microseconds. This timestamp delay is significantly smaller than the rate in seconds at which the sensor stations will be receiving the tag signals. Thus, this time delay due to calling the timestamp function was deemed acceptable for the scope of this application.

2.2.3 UART Communication

In order for the data received from the tag to be relayed through the mesh network subsystem, a universal asynchronous receiver / transmitter (UART) communication protocol became necessary. The sensor station became divided into two CC1310 microcontroller units connected with a UART serial communication protocol. One CC1310 MCU would serve as the receiver to receive all data transmitted by surrounding tags, and the other CC1310 MCU would serve as a transmitter and part of the mesh network subsystem to relay data received on a UART channel wirelessly through the mesh network protocol.

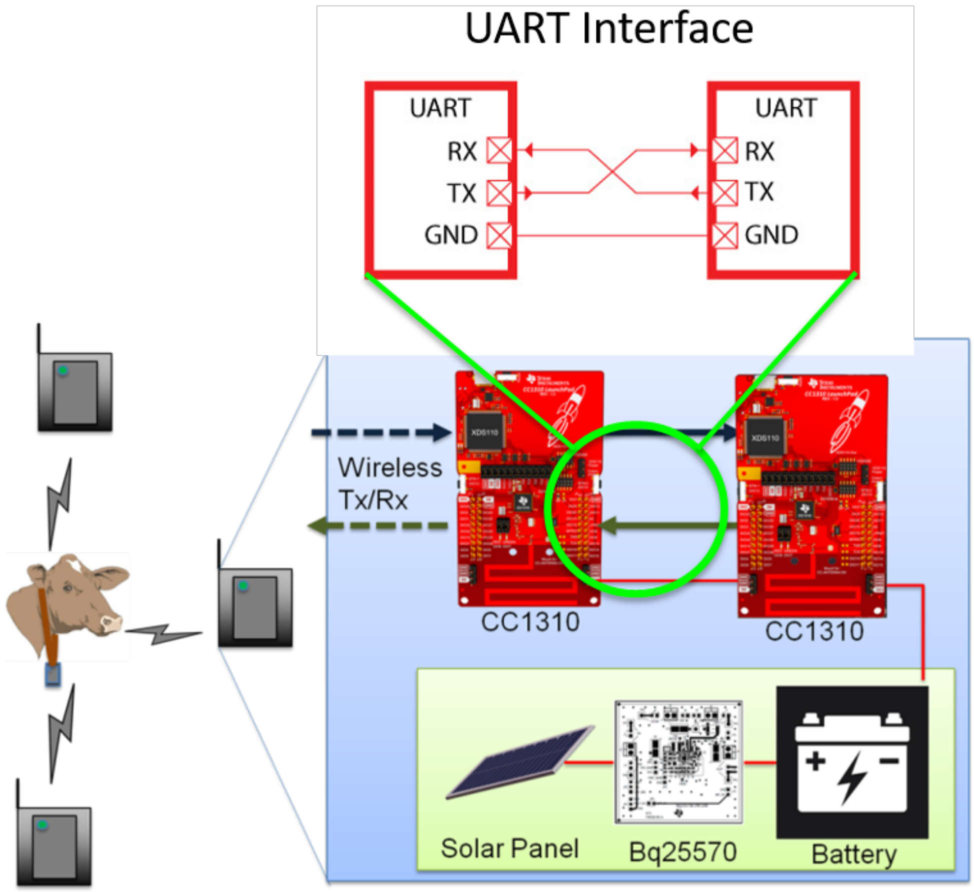


Figure 4: Two CC1310 MCUs at Sensor Station with UART Communication

The UART communication was developed using the UART driver libraries in the embedded tools ecosystem, TI Real Time Operating System (TI-RTOS). The UART communication protocol is able to read and write serial data to the serial Tx and Rx pins on the CC1310 MCU.

In order to comply with the mesh network specifications, several adjustments was made to the UART communication protocol between the two CC1310 MCUs. The first is setting the baud rate of the UART connection at 115200 bits per second. The second is enabling the UART_RETURN_FULL parameter so that when the buffer is full, a read action is unblocked and returns a newline character. The newline character ending the string of data from the tag is used in the mesh network protocol to indicate when to transmit the buffer of data through the mesh network. The third setting was to use sequence number 8 in hexadecimal at the beginning of all data transmitted by the tag such that consistent transmission of data would be possible across the mesh network. After these adjustments were made, all data relayed by the tag was able to be transmitted through the mesh network protocol.

2.2.4 Detect Received RSSI at Sensor Station

The functionality of being able to detect the received signal strength index (RSSI) at the sensor station became necessary when timestamping of received tag data was deemed too imprecise for triangulation and data analytics. RSSI measured from a signal sent by a tag would be sent along with the received tag data through the mesh network to assist in triangulation of the tag's location relative to three nearby sensor stations.

RSSI detection was implemented using the CC1310 EasyLink application program interface (API) layer on top of the base CC1310 RF Driver. An RSSI library included in the EasyLink API was used to estimate the RSSI of an incoming signal received at the sensor station.

The precision of the RSSI estimations became of concern as they would have an integral role in the triangulation calculations performed at the base station. A few tests were performed to validate the precision of the RSSI estimations by comparing two RSSI measurement programs, TI's SmartRF Studio, and the EasyLink RSSI measurements. The results of these tests are shown in Table 4.

RSSI Validation

Location	SmartRF Studio (dB)	EasyLink (dB)	Difference (dB)
1	-29.225	-30.75	**-1.525**
2	-49.025	-53.7	**-4.675**
3	-68.9	-66	**-2.9**
4	-54.25	-59	**-4.75**
5	-69.925	-67.8	**-2.125**

Table 4: RSSI Precision Validation

In Table 4, the RSSI values measured by SmartRF Studio and EasyLink in dB are recorded for 5 different locations. The difference in dB between these two measurements is then calculated and presented in the table. It can be noted that a maximum difference of roughly 4.6 dB was made between the two RSSI measurements. In the data analytics subsystem, it would later be shown that this difference in RSSI precision does little to affect the noisy triangulation calculations. Thus, RSSI measurements are able to be measured at the sensor station and relayed through the mesh network along with tag information data.

3 VALIDATION PLAN

In the validation plan of the Tag and Sensor subsystem, the objectives that were sought to be met were:

- Test power consumption of tag processes by measuring the current drawn by the tag on an average application. Ensure that these results allow the tag to operate at a minimum of 2 years when powered by AAA batteries.
- Program the sensor stations to timestamp a signal to aid in triangulation computations. The timestamp values should be accurate to a second of reception.
- Test power consumption of sensor processes by measuring the current drawn by the sensor on an average application. Ensure that the sensor power consumption does not exceed the recharging capabilities of the solar panel power subsystem linked up to the sensor stations.
- Sensor stations should detect the average received signal strength indicator (RSSI) in dB within 5 dBs.
- Sensor stations should accurately transmit data through UART at a baud rate of 115200 bits per second.

The five objectives set for the Tag and Sensor Subsystem were all attempted throughout the semester and the subsystem performances regarding each objective was tested to determine whether or not the set criteria were met.

In regards to the power consumption of the tag processes, the average current drawn by a tag component over 5 seconds of operation was computed to be 0.4894 mA. With this amount of current being drawn over 5 seconds of operation and the tag being powered by a coin-cell battery, it was estimated that the tag might be able to run for about 1 year. This is below the threshold set of a minimum of 2 years. Thus, the protocol for power conservation will be reevaluated for the tag to improve its power efficiency and ensure a longer battery life.

A proposed solution to improving the power conservation of the tags is to reduce the time at which the pulse of information is transmitted by the tags. By reducing the time over which the pulse is sent at, the current being drawn over 5 seconds can be reduced. Alternatively, further research into low-power modes on the CC1310 microcontroller unit can lead to a solution to this issue.

When evaluating the results of timestamping a received signal at the sensor station, it was determined that the CC1310 MCU could timestamp a signal within 1 second of precision. Much more precise results could be obtained but for this application, 1 second precision was satisfactory for the validation plan.

Evaluating the results of the tests of the wake-on radio mode for the sensor stations, it was determined that the average current drawn by a sensor station over five seconds of operation is 0.268 mA. This is satisfactory as it does not exceed the recharging capabilities of the solar panel as described in the power subsystem and allows for the lithium-ion battery to properly power the sensor station module and be recharged during the day cycle.

Evaluating the precision at which the sensor stations were able to detect the received signal strength indicator demonstrated that the program developed was able to measure received signal strength index at a precision of roughly 4.6 dB of error. This is within the 5 dB of acceptable difference for use in noisy triangulation in the data analytics subsystem.

The sensor stations were able to accurately transmit data through UART at a rate of 115200 bits per second with the enabling of two different features: adding a newline character at the end of a full buffer and adding a sequence number of 8 in hexadecimal at the beginning of the transmitted data. Once these additions were made, the data was able to be transmitted without error and consistently through the mesh network subsystem.

SOFTWARE AND DATA ANALYSIS SUBSYSTEM

Randy Ardywibowo

1. Introduction to Software and Data Analysis Subsystem

The following document is a representation of the software subsystem. This subsystem consists of a back-end data analysis system, and a front-end which shows cattle position in a Graphical User Interface (GUI). The back-end's operation consists of receiving distance data from the mesh network and cattle tagging system, and performs trilateration to reconcile the data into cattle position. The front-end system displays the analyze data and display the position of each cattle on a GUI. This GUI also displays relevant information pertinent to each cattle and their system information such as their name or identification, recorded weight and size, and the status of their battery level.

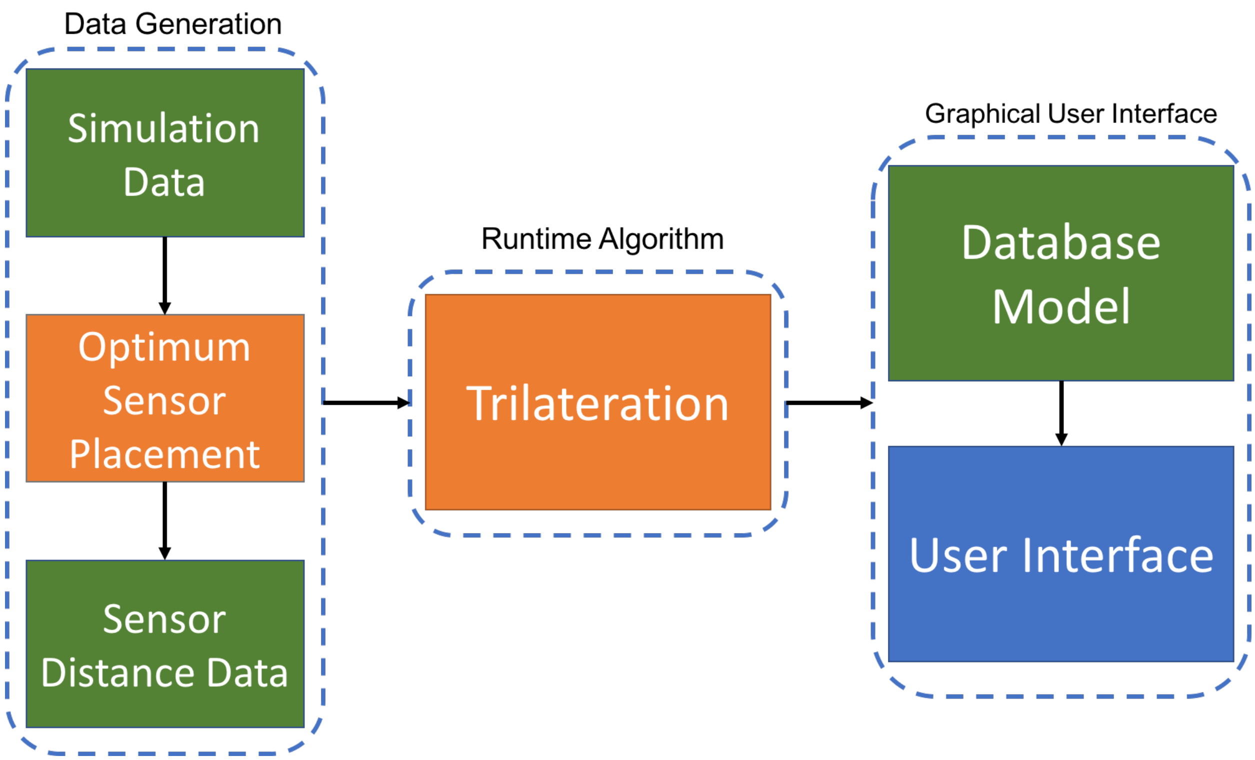


Figure 1: Development phase and subtask relationship diagram of the software subsystem

1.1 Concept of Operation

The concept of operation of the Software Subsystem is to accumulate the data transmitted by the mesh network subsystem and reconcile it as cattle position data in a graphical user interface (GUI). The system will perform trilateration of the received distance measurements and display data visualizations useful to the rancher. This includes battery warnings, cattle outlier detection, and heat maps of cattle location tendency.

1.2 Functional System Requirements

The Functional System Requirements document further describe the requirements for the software subsystem. It describes that the system should be able to track at least 24 tags simultaneously. This requires efficient position estimation algorithms, which is described further in this section. Moreover, tracking accuracy of the position estimation should be within 2 meters of accuracy. Depending on the severity of the noise, the tracking accuracy will depend on the amount of redundancy put in the sensor placement. Finally, the system must also inform its user of any communication failure within the sensor communication system. As it is a matter of integration, this feature will be incorporated in the integration phase of the product's development.

1.3 Interface Control Document

The Interface Control Document further describes the function of the software subsystem. The software subsystem shall give tag battery level notifications, heat maps of herding location, tag inactivity warnings, as well as tag out of bounds warnings. Some of which have been implemented this semester while others will be implemented as part of the integration plan for future project plans. These features will be discussed in detail in the following sections.

1.4 Development of Subsystem

1.4.1 Simulation Data Generation

To begin development of the software subsystem, simulation data must first be gathered to demonstrate the data analysis methods properly. The simulation data should consist of a reasonable number of cattle whose movement accurately displays herd movement behavior. This data should also contain sufficiently dense time points over a good range of time to simulate the system under high data throughput conditions. Because of this, the specification for the simulation data should be that it is a simulation of herd behavior of at least 100 cattle, with measurements taken at least once every 30 minutes, and that the data should span at least 1 year.

To accomplish this, cattle movement data was gained from the Starkey Project [1]. This project studies the movement of animals such as deer, elk, sheep, and cows over a period of 5 years. From this dataset, over 120 cows were observed which satisfies our condition on the number of cows required. However, the density in which the cattle positions were sampled was not sufficient, as measurements were taken once every 5 days. To fix this, a mean reverting random-walk algorithm was fitted. This algorithm, called the Ornstein-Uhlenbeck (OU) process, is a mean reverting extension of the Wiener process, the standard random walk algorithm. It is a Stochastic Differential Equation (SDE) which is a summation of a drift term and a diffusion term. In 2 dimensions, this system is represented by the following equation:

$$\begin{bmatrix} dX(t) \\ dY(t) \end{bmatrix} = \begin{bmatrix} \mu_x\{\mathbf{r}(t), t\} \\ \mu_y\{\mathbf{r}(t), t\} \end{bmatrix} \mathrm{d}t + \mathbf{D}\{\mathbf{r}(t), t\} \begin{bmatrix} d\mathbf{\Psi}_x(t) \\ d\mathbf{\Psi}_y(t) \end{bmatrix}$$

Here, X and Y represent the X and Y positions of each cattle, μ_x and μ_y representing the drift parameter, a parameter which sets the mean of the cattle's position, D, the diffusion matrix, describes the correlation between steps in the x and y direction, ψ_x and ψ_y are the Wiener random processes with zero mean.

Using this equation to optimize the simulated data with the data given, the drift term was estimated by approximating the SDE by the difference equations:

$$(X_{i+1} - X_i)/(t_{i+1} - t_i) = \mu_x(X_i, Y_i, t_i) + \sigma_x \varepsilon_{1i}/\sqrt{t_{i+1} - t_i}$$
$$(Y_{i+1} - Y_i)/(t_{i+1} - t_i) = \mu_y(X_i, Y_i, t_i) + \sigma_y \varepsilon_{2i}/\sqrt{t_{i+1} - t_i}$$

This model assumes a diagonal diffusion matrix, where the X position change doesn't correlate with the Y position change. This is partly justified by work done by Preisler et al., who have shown that there is negligible correlation between the change in X and Y positions within this data set [2]. This means that each X and Y of the diffusion term can be broken down into a 1-Dimensional case of the OU process respectively. This process is show in the bottom Figure.

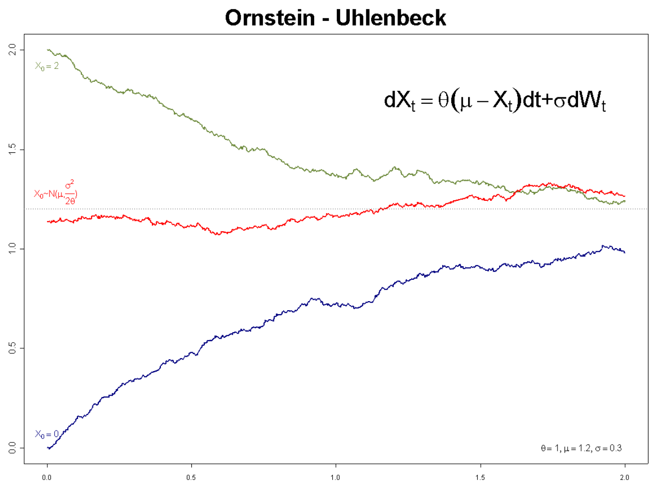
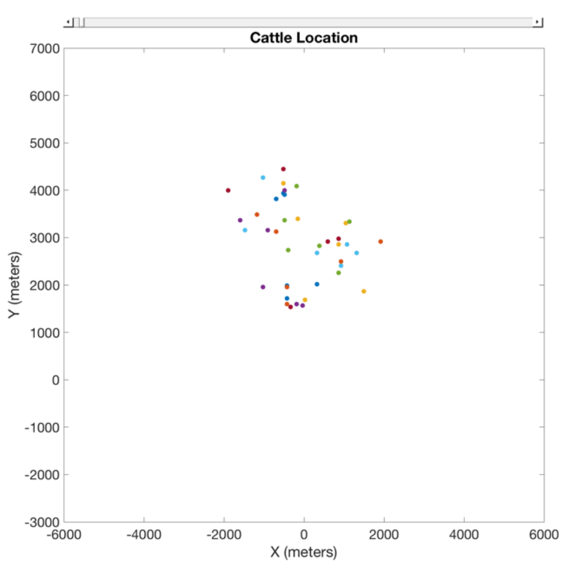
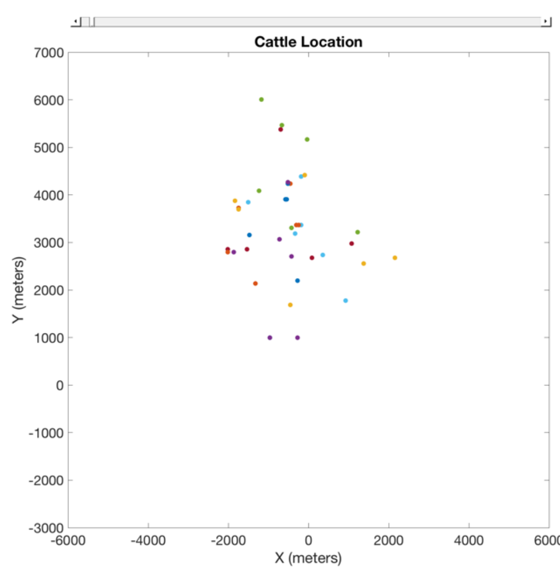


Figure 2: 1-Dimensional case of the OU Process

To implement this model, we use available code from Preisler et. al., who has studied this system to predict animal movements based on previous observations [2]. A frame of the simulated cattle movement data is shown below:





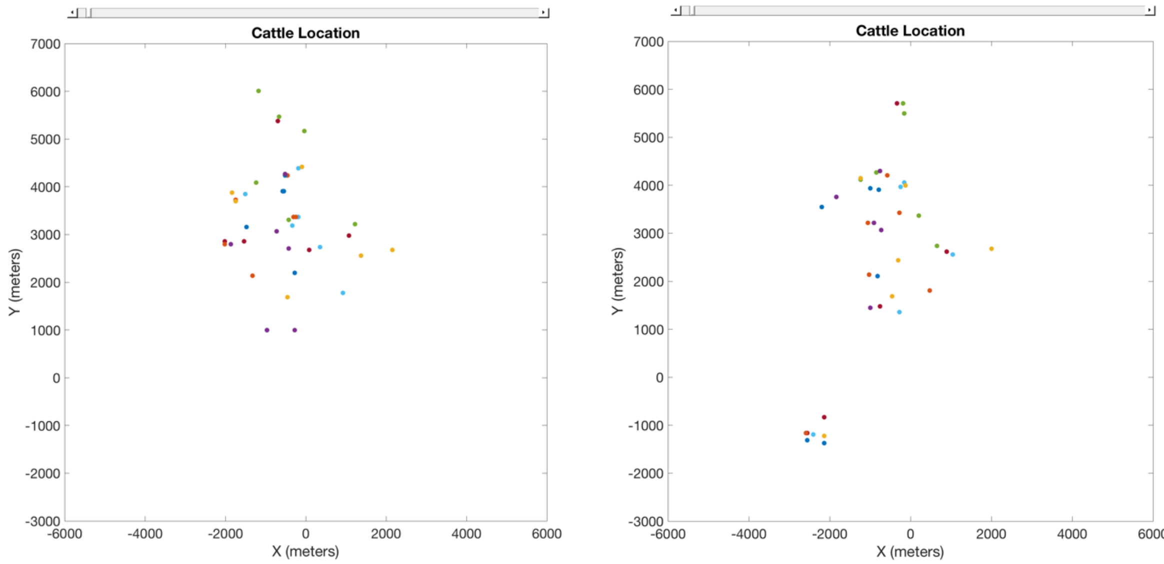


Figure 3: Cattle movement simulation

1.4.2 Optimum Sensor Positioning

Because the data generate is in the form of distance data and not in X and Y coordinates, the data gained from the simulation must be transformed to the particular sensor placement scheme. Because of this, study in the optimal sensor positioning must be done so that cattle position can be reconciliated with a minimal amount of sensors. Thus, as trilateration requires a minimum of 3 sensors to cover a given area. We require a 3-coverage method to produce optimal sensor placement.

To do this, we used a 3-coverage method following the work done by Kim et al [3]. This approach combines together three layers of sensors, each having optimal coverage on their own. With R as the effective radius of each sensor, each layer has sensors placed at $\sqrt{3}R$ appart from each other. The three sensor layers are shown on the figure below:

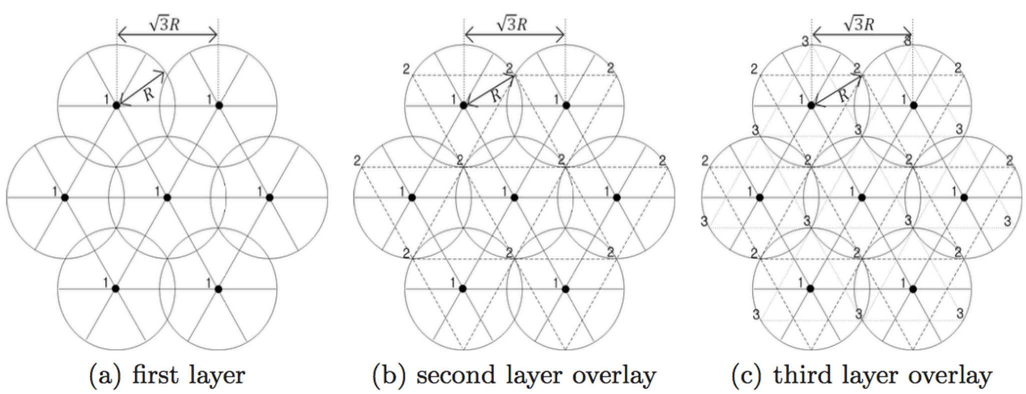


Figure 4: 3-Coverage sensor placement

We implemented this system in MATLAB and produced simulation data of distance measurements according to it. From our simulation, with the sensor range that we have currently, 2200 sensors to cover a 14 x 7 ≈ 100 km^2 area. This is effectively optimal, as the upper bound of the amount of coverage of three sensors is ¾ of their area. This means three sensors accounts for about 0.049 km^2. Thus, the upper bound of the area covered by 2200 sensors is 108 km^2. However, accounting for overlap and redundancy of the area covered, as well as the redundant coverage of the sensors placed at the edge, this coverage is optimal. The results of the placement are shown below.

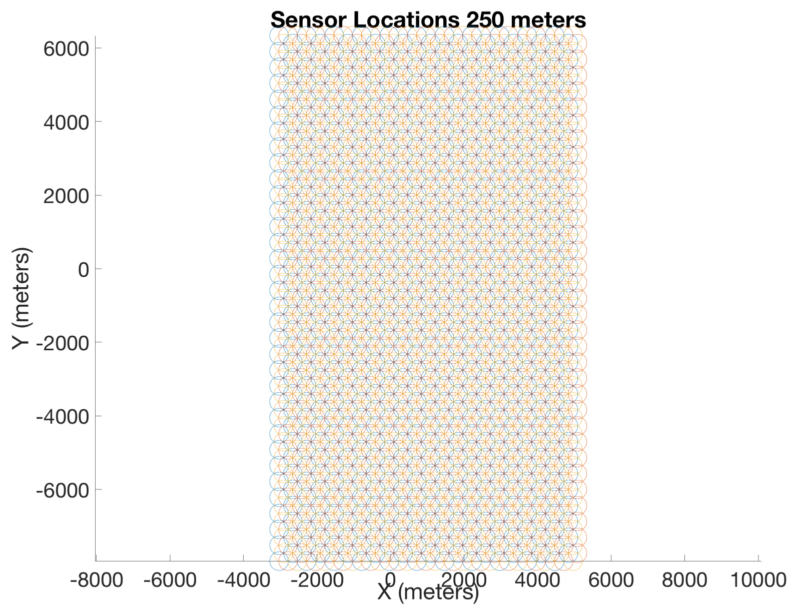


Figure 5: 3-Coverage of Sensors with range 250 meters

A second simulation was done with an effective sensor range of 2000 meters. With this range, we can cover the same region using only 50 sensors. This sensor arrangement is shown in the figure below.

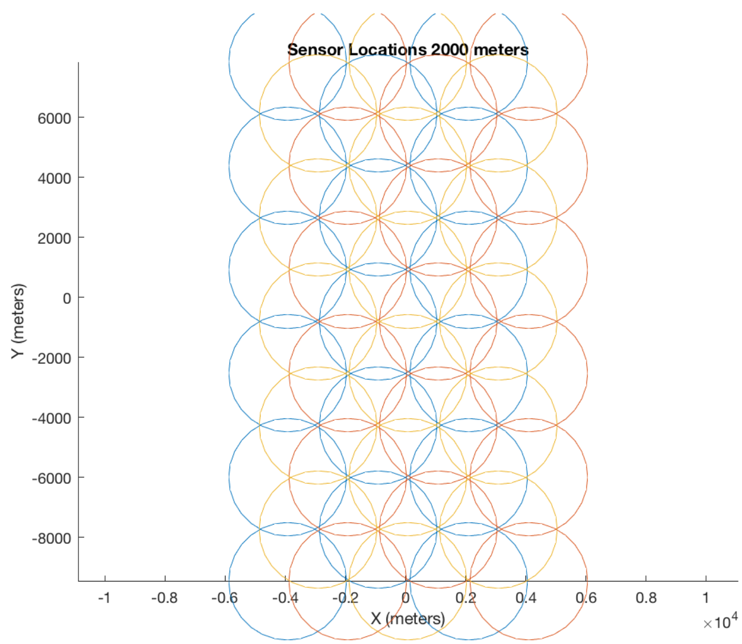


Figure 6: 3-Coverage of sensors with range 2000 meters

1.4.3 Trilateration under Noisy Distance Measurement Data

Using the data generated, the trilateration algorithm can now be developed. This simulation, needs to consider possible noise in the distance readings as in working conditions, as pinpoint accurate data is not possible in working conditions. Because of this, the trilateration algorithm we desire is a system which is robust to noise. To do this, the trilateration algorithm must be such that it minimizes the error of the following circle equations:

$$\left(x - x_p\right)^2 + \left(y - y_p\right)^2 = (r + \varepsilon)^2$$

Here, x and y are the position of each sensor while x_p and y_p are the estimate of the position of each cattle. On the other hand, r is the true distance measurement gained from the sensor while ε is the additive error which we model as a normal random variable with mean zero and variance σ^2. Denoting the number of sensors that senses a cattle's distance as N, our optimization problem becomes the following:

$$\min_{x_p, y_p} \sum_{n=1}^{N} \left\| \left(x_n - x_p\right)^2 + \left(y_n - y_p\right)^2 - r_n{}^2 \right\|$$

This corresponds to minimizing the sum of the Euclidean norm of the errors ε_n. This gives us a least squares estimator to the position of each cattle. This system will also integrate redundant distance measurement data in addition to handling errors, improving the cattle position estimation further.

As shown in the figure below, this optimization problem is convex and nonlinear, and we can use convex optimization strategies such as gradient descent on the problem. Here we used the Levenberg-Marquardt algorithm. This optimization algorithm solves non-linear least squares problem and is a variant of the gradient descent algorithm [4].

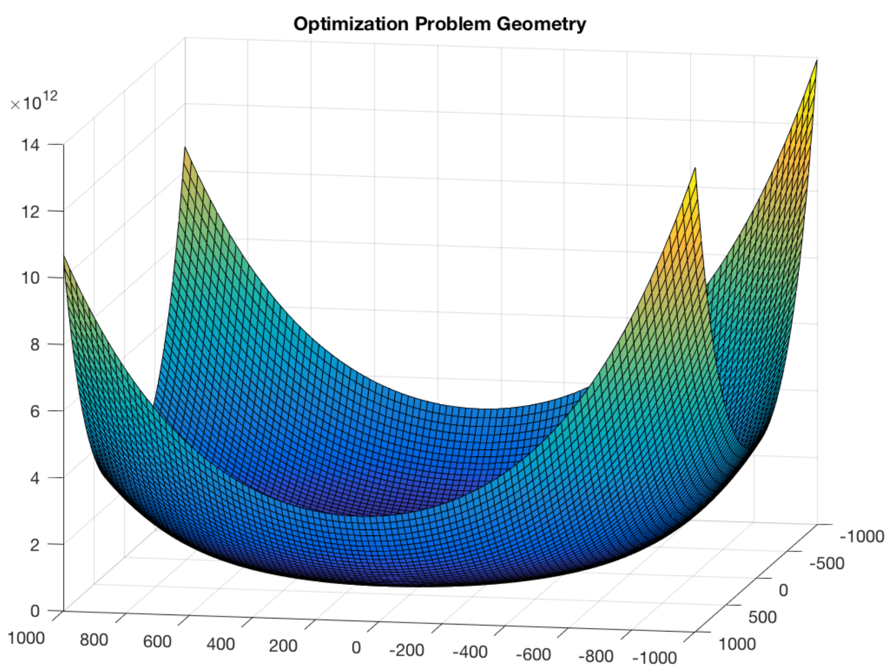


Figure 7: Convexity of the optimization problem

The algorithm was implemented in MATLAB and a single result of trilateration is shown below. Here the algorithm can estimate the cattle's position even when there is no clear intersection of the three circles.

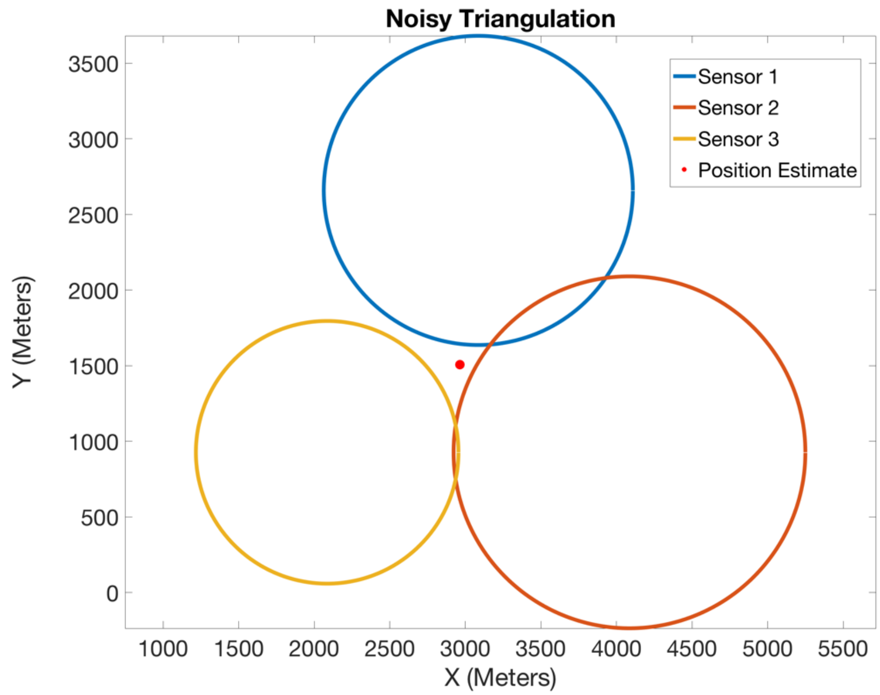


Figure 8: Demonstration of the trilateration algorithm

1.4.4 GUI Development

The GUI was developed using Meteor, an application development framework for web and mobile devices. Using Electron, the web application can be ported to the desktop fully featured without any issues in additional development. Front end development was done using Angular 2. Made by Google, this is an open-source web application framework which imposes a model-view-controller (MVC) architecture on its application. The application is written using TypeScript, a superset of JavaScript developed by Microsoft which imposes type definitions to JavaScript objects. For the user interface, we used a Material Design layout provided by Google. This allows us to quickly develop beautiful user interfaces by using prebuilt elements from the Material Design library. Finally, MongoDB was used as a NoSQL database service to save and load data in the application. Multiple other libraries were also used in the creation of the GUI. A complete list of these libraries is shown below:

Library Name	Usage
Angular	Base front-end framework
BCrypt	Encryption service for data protection
D3	Plotting and graphical object manipulation
Electron	Desktop application port
Google Maps	Map display service
Ionic	Mobile application components
Material	User interface components

Meteor	Base application framework
Mongo	Database framework
NPM	Library and package manager
Papaparse	Parsing of CSV files
RxJS	Asynchronous processing and database queries

Table 1: Frameworks and libraries used in the GUI

1.4.5 Program Structure

The program is structured in a Model-View-Controller (MVC) architecture. This architecture provides sufficient organization of code as code is divided into UI elements, UI controllers, and underlying models and database constructs. This architecture is shown below:

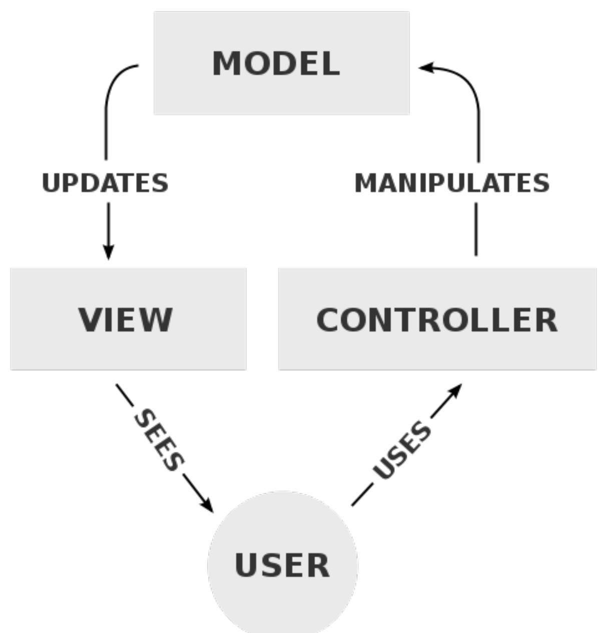


Figure 9: Structure of the MVC architecture

Using this structure, the application is divided into a main model which stores all the data relevant to each cattle, two views for the cattle list and map respectively, as well as controllers for each. The cattle list view controller is subsequently divided into two more views and controllers, a controller for the list of cattle and a view for searching the list based on the cattle's name. This structure is shown in the following figure:

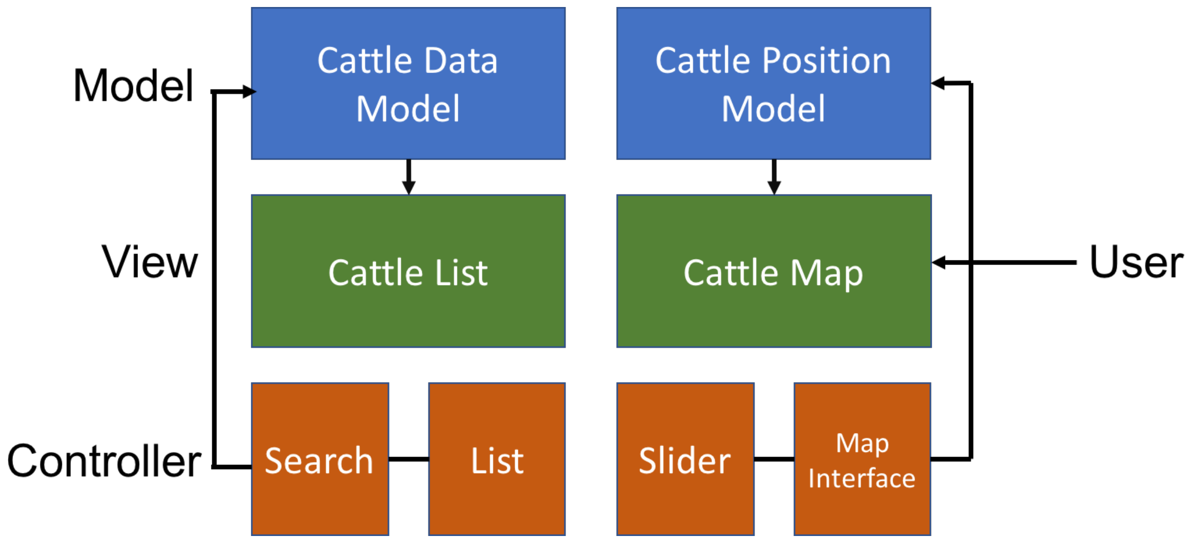


Figure 10: MVC Layout of the Ranch Hand GUI

1.4.6 Google Maps Integration

To display the cattle position in the GUI meaningfully, we elected to use the Google Maps API to provide an overlay to the cattle position data. This requires that the data be converted to latitude and longitude to input to the map API.

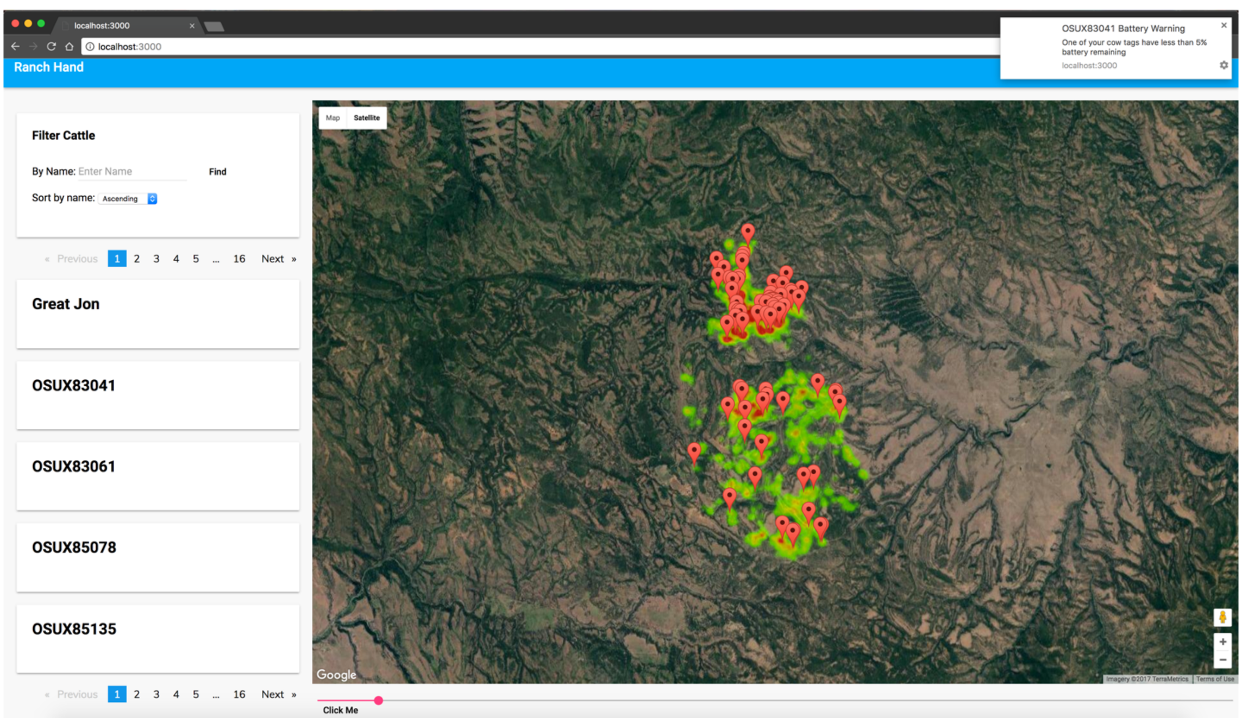


Figure 11: Screenshot of the Ranch Hand user interface

2. Collected Data and Validation

2.1 Accuracy of the Simulation Data Generated

Because the data generated served to impute missing values in between measurements, no good error measurement can be gained as the observed data equals the generated data in the time points where the observed data is available. However, Preisler et al. studied this model for use with prediction and found the locations predicted had a mean error of 53m [2].

2.2 Optimality of the Sensor Placement

From our simulation, with the sensor range that we have currently, 2200 sensors to cover a 14 x 7 ≈ 100 km^2 area. This is effectively optimal, as the upper bound of the amount of coverage of three sensors is ¾ of their area. This means three sensors accounts for about 0.049 km^2. Thus, the upper bound of the area covered by 2200 sensors is 108 km^2. However, accounting for overlap and redundancy of the area covered, as well as the redundant coverage of the sensors placed at the edge, this coverage is optimal.

2.3 Sensor Amount for Sufficient Coverage

Based on the sensor placement results using a sensor range of 250 meters, more than 2000 sensors are required to cover a 100 km^2 area. This is not sufficient based on our functional system requirements, as it requires at least a 1 km sensor range which would correspond to about 170 sensors to cover a 100 km^2 area. This needs to be improved by the corresponding subsystems. Our results show that for our system to be feasible, an antenna must be added to the sensor systems to increase its range of coverage to the sufficient requirement. The relation between sensor range and amount of sensors is shown below:

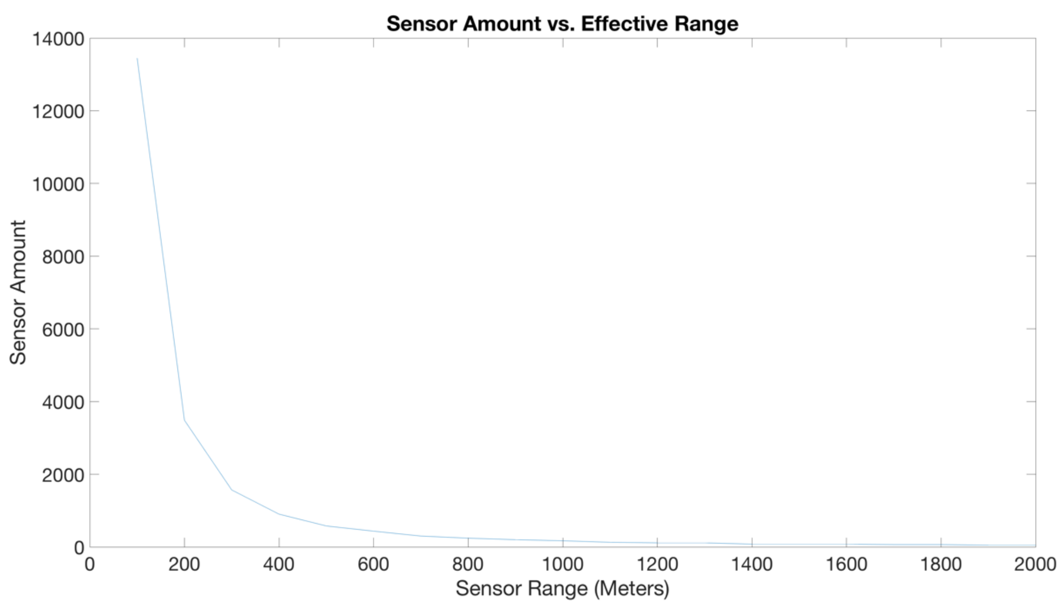


Figure 12: Sensor amount vs. effective range of optimal sensor placement

2.4 Trilateration Mean Squared Error Simulation Measurement

The accuracy of the trilateration method was measured with different variances of Gaussian noise added. From the simulation, we found that the Mean Squared Error of the different measurements equal to the standard deviation of the Gaussian noise given. This is optimal, as with the current sensor configuration favoring coverage compared to redundancy, the best estimate of the cattle's position should have mean square error value equal to the noise's standard deviation. This is because the minimum number of sensors required for triangulation is three sensors. With this amount, the trilateration algorithm can't serve to correct for errors as there is no redundancy added to the data collection.

The sufficiency of this method will depend on what type of error we receive from the other subsystems. However, if no better error variances can be gained from the sensor tag system we currently have, additional redundancy in the sensor placement can be added. This would mean that more sensors would need to be added to cover a smaller space, which would modify the sensor placement calculation by lowering the effective range of each sensor.

2.5 Speed of Trilateration Optimization

The speed of position estimation through trilateration depends on how accurate of an estimate wanted. For an accuracy of 2 meters specified in the Functional System Requirements (FSR), this calculation was measured to take no longer than 0.04 seconds with a computer running an i7-4790k processor with no dedicated graphics acceleration. The time complexity of performing trilateration is equivalent to the time complexity of performing a pseudo-inverse, an operation to calculate the least squares optimization. This has a time complexity of $O(n^3)$, with n being the number of columns/rows of a matrix. Although it is polynomial time, for our problem, n will be the number of equations in our optimization which is on average 3. This is sufficient for our system as it will be able to run on a regular desktop computer in near real-time with the given complexity.

2.6 Responsiveness of the Graphical User Interface

The graphical user interface was designed such that any long running tasks such as querying a database is done asynchronously through the RxJS framework. This allows long tasks such as triangulation to be done in the background while the front-facing user interface remains reactive of functional. Although no true measure of the responsiveness of the application can be gained without proper user testing, this factor has been kept into consideration throughout the development process.

3. Validation Plan

Sensor Placement and Data Generation Validation Plan:

1. Compare data of simulated cow movements with original un-imputed data.
2. Record sensor amount of various sensor ranges.
3. Compare sensor coverage with upper bound sensor coverage.

Trilateration Validation Plan:

1. Measure the mean squared error of position estimate with true position of cattle over numerous measurements.
2. Compare them with respect to the standard deviation of the noise generated.

Graphical User Interface Validation Plan:

1. Verify that the GUI is functioning as intended without any bugs in the software.
2. Test position estimation on the GUI program to verify that it matches with the simulation.

References

[1] "Welcome to the Starkey Project." *Starkey Project*. N.p., n.d. Web. 13 Oct. 2016.
[2] Preisler, Haiganoush K., et al. "Modeling animal movements using stochastic differential equations." *Environmetrics* 15.7 (2004): 643-657.
[3] Kim, Jung-Eun, et al. "Sensor placement for 3-coverage with minimum separation requirements." *International Conference on Distributed Computing in Sensor Systems*. Springer Berlin Heidelberg, 2008.
[4] Gill, Philip E., and Walter Murray. "Algorithms for the solution of the nonlinear least-squares problem." *SIAM Journal on Numerical Analysis* 15.5 (1978): 977-992.

APPENDIX A ACRONYMS AND ABBREVIATIONS

Below is a list of acronyms used in this document:

BIT	Built-In Test
GUI	Graphical User Interface
Hz	Hertz
ICD	Interface Control Document
kHz	Kilohertz (1,000 Hz)
mA	Milliamp
MCU	Microcontroller unit
MHz	Megahertz (1,000,000 Hz)
MVC	Model-View-Controller
mW	Milliwatt
OU	Ornstein-Uhlenbeck
PCB	Printed Circuit Board
PC	Personal Computer
RMS	Root Mean Square
SDE	Stochastic Differential Equation
TBD	To Be Determined
UART	Universal asynchronous receiver / transmitter

APPENDIX B DEFINITION OF TERMS

No terms needed to be defined within the document.