Hacking RFID Tags to Produce Economical Soil Moisture Sensors

by
Brett S. Stoddard

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Electrical and Computer Engineering
(Honors Scholar)

Presented June 4, 2019
Commencement June 2019

AN ABSTRACT OF THE THESIS OF

Brett Stoddard for the degree of Honors Baccalaureate of Science in Electrical and Computer Engineering presented on June 4, 2019. Title: Hacking RFID Tags to Produce Economical Soil Moisture Sensors.

Abstract approved:_____

Chester Udell III

Currently available soil moisture sensors have intrinsic limitations that limit their adoption by industry. Depending on the technology, these limitations may include high costs, battery requirements, precice manual installation, high cost, and/or constrictive legislation that regulates their use. A passive soil moisture sensing system based around RFID technology would allay many of these issues. Moisture sensing RFID tags are already being used in other industries and they present a unique opportunity within the agricultural space, such as Smartrac's Dogbone RFID Tag. Today, Dogbone tags inexpensive, don't require batteries, charging, recalibration or maintenance. In this thesis, those tags were adapted and calibrated to sense the moisture content of soil. The result was an affordable, scalable, and highly granular system for sensing volumetric water content (VWC) in-situ. Calibration was executed using an industry-standard probe in reference soil at a range of moisture values soil ranging from 9-31% VWC. This validated a definite relationship between tag sensor values and the VWC However, some concerning outliers were recorded. Further testing examined the potential of using these tags with drones, on a center pivot, and integrated with an app.

Key Words: RFID, Soil Moisture, Calibration,

Corresponding e-mail address: stoddardbrett@gmail.com

Hacking RFID Tags to Produce Economical Soil Moisture Sensors

by
Brett S. Stoddard

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Electrical and Computer Engineering
(Honors Scholar)

Presented June 4, 2019
Commencement June 2019

Honors Baccalaureate of Science in Electrical and Computer Engineering project of Brett Stoddard presented on June 4, 2019.

APPROVED:

_____

Chester Udell III, Mentor, representing Department of Biological and Ecological Engineering

_____

John Selker, Committee Member, representing Department of Biological and Ecological Engineering

_____

Matthew Johnston, Committee Member, representing Department of Electrical Engineering and Computer Science

_____

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, Honors College. My signature below authorizes release of my project to any reader upon request.

_____

Brett Stoddard, Author

Hacking RFID Tags to Produce Economical Soil Moisture Sensors

Brett S. Stoddard

I. Abstract

Currently available soil moisture sensors have intrinsic limitations that limit their adoption by industry. Depending on the technology, these limitations may include high costs, battery requirements, precice manual installation, high cost, and/or constrictive legislation that regulates their use. A passive soil moisture sensing system based around RFID technology would allay many of these issues. Moisture sensing RFID tags are already being used in other industries and they present a unique opportunity within the agricultural space, such as Smartracs Dogbone RFID Tag. Dogbone tags cost cents per tag, and dont require batteries, charging, or maintenance [1]. In this thesis, those tags were adapted and calibrated to sense the moisture content of soil. The result was an affordable, scalable, and highly granular system for sensing volumetric water content (VWC) in-situ. Calibration was executed using an industry-standard probe in reference soil at a range of moisture values soil ranging from 1-31% VWC. This validated a definite relationship between tag sensor values and the VWC. Further testing examined the potential of using these tags with drones, on a center pivot, and integrated with an app.

II. Introduction

In recent decades, there has been an increasing attention to the limited availability of freshwater. The largest consumer of freshwater in the United States is agricultureaccounting for 60-90 percent of freshwater withdrawals in the US, depending on region [2][3]. Therefore, a small increase in irrigation efficiency would have an noticeable positive impact on usage.

One possible improvement comes in the form of a system to track subsurface fluid flow–a process that is highly heterogeneous and as difficult to model. This suggests that empirical method for measuring soil moisture could be beneficial. If such a system existed, decisions regarding the amount and frequency of irrigation from location to location could be made with higher accuracy, thus increasing irrigation efficiency. However, barriers persist that have prevented widespread adoption of such a system. One reason for this is that current price of soil moisture sensors renders such a system uneconomical. However, the RFID soil moisture sensors examined in this thesis can be manufactured and sold at significantly lower cost than available products.

A. Soil Moisture Sensors

One of the most accepted units of soil moisture is volumetric water content (VWC). The most accurate and practiced method for measuring VWC is a laboratory test in which a known volume of soil is weighed before and after it is dried [4]. This is a labor intensive, and destructive test which makes it uneconomical at agricultural scales. Another employed method for measuring the volumetric water content of soil involve neutron scattering, which is nondestructive and has the potential to be automated, but requires recalibration, is expensive, and is regulated which limits its agricultural uses [4]. The two most commonly used soil moisture measuring techniques are resistivity sensors, and frequency domain probes (FDP) or time domain reflectivity probes (TDP)[4][5]. These measure water content indirectly through the electrical properties of the soil in the very high frequency (VHF) and ultra high frequency (UHF) ranges which highly correlate with VWC [6].

Satellites, such as NASAs Soil Moisture Active Passive (SMAP) and ESAs Soil Moisture and Ocean Salinity (SMOS), collect region scale maps of surface soil moisture content. One goal of these missions is to benefit agricultural production models and industry overall, and they have some benefit to individual farmers but are limited by low resolution [7][8]. A potential hybrid system could be created that combines an individual soil moisture sensor at one or multiple known problem areas within a field with data from SMAP or SMOL. This could provides an average to come up with a more cohesive understanding of the current VWC however, the explicit proposal of such a system is beyond the scope of this thesis. It should be noted that in 2015 SMAP suffered a catastrophic failure that crippled the radar (or active) portion of its instrumentation but is still operational in a limited capacity[9].

B. RFID Sensors

Since its patenting by inventor Charles Walton in 1980, passive radio frequency identification (RFID) technology has found use in multiple industries with a diverse range of applications[10]. Passive RFID systems have been used in everything from inventory tracking to building security measures. The low cost to produce a tag, the small form factor, the ability to be read through nonconductive material, and ability to function without a battery been primary reasons for their widespread adoption. At its most basic implementation, RFID technology allows for reading a set of pre-programmed identification bits by way of a magnetically coupled circuit between the tag and an external antenna [10].

Recent advances in the technology has allowed RFID tags to sense their environment. Tags that can sense moisture, presence of certain chemicals, and temperature have been proposed or demonstrated [11][12][13][14]. Furthermore, RFID soil moisture sensor have been proposed and demonstrated with some success [15][16][17][18][19]. Most of these attempts

to create a RFID-based soil sensor face several shortcomings which our proposed design avoids.

The two most common pitfalls for soil moisture RFID implementations are if they use signal strength to detect soil moisture, or are battery powered (actively powered instead of passive). RFID systems to sense soil moisture have been attempted in the past by using Received Signal Strength Indicator (RSSI) as values to predict VWC [16][15]. RSSI is a measure of reflected from the tag to the to the RFID receiver in units of decibels (dB). This method for measuring VWC is promising in laboratory conditions, however, it is infeasible in-situ as it is extremely sensitive to extraneous factors such as distance between the sensor and receiver. Also, it is sensitive to noise on the integrated path between antenna and tag [18][16].

Other commercially available RFID tags, known as active RFID, use batteries to power their circuitry [19]. This technology benefits from substantially longer read ranges than our proposed system. However, powered RFID tags require maintenance in the form of maintaining a charge in said power source. Additionaly, powered RFID tags are commonly built from printed circuit boards (PCB) which have a worse environmental impact than our proposed tags [20].

In 2012, companies RFMicron and SmarTrac announced a partnership line of integrated circuit chips with Chameleon technology [21]. Chips with this "Chameleon Engine" have an embedded dynamic capacitor bank that works to retune the signal when the tag is in an high dielectric environment. A sensor code can be read from an RFID antenna that describes how much of the capacitor bank was activated during tuning. Because water has a high relative permittivity constant, RFID tags are sensitive to the moisture content in their environment and can be used as sensors [22].

In 2015, RFID tag manufacturer SmarTrac started selling these RFID moisture sensors. Because they sense local capacitance rather than RSSI or minimum activation energy, they are more practical for in situ measurements. The Dogbone tag was developed during this partnership primarily for the product storage industry. It was chosen to be used in our testing because its default sensitivity allows it to accurately read over a large range of moisture levels rather. Many other tags are too sensitive to be useful or return boolean wet or dry values [22].

Figure 3 highlights the water sensitive area of a Dogbone tag. When water is near this patch, the relative permittivity of the environment surrounding the tag increases due to waters high relative permittivity [23]. By using commercially available tags, time-to-market can be reduced, as well, proof-of-concept has already been established in other industries [24].

III. Design

Preliminary testing with the Dogbone tags revealed several important factors that interfered with the validity of readings. These included the material that the tags were mounted onto and if the tags were deformed (i.e. bent). Smartracs white paper on the RFID tags specifies that the tags were specially designed to be attached to and operate on metal surfaces

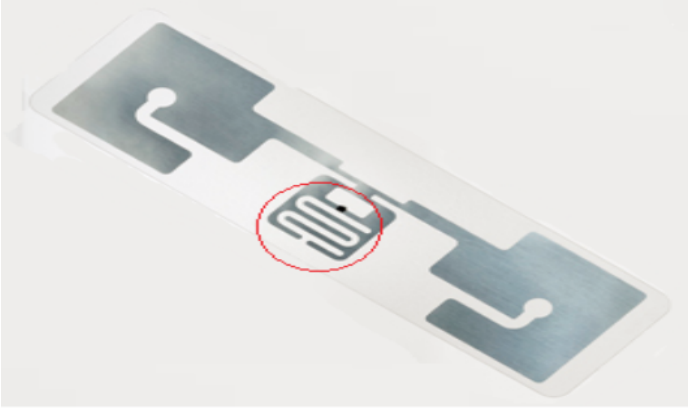

Fig. 1: Image of Dogbone RFID tag with the moisture sensitive area circled in red

	Tag A Metal	Tag B Metal	Tag A Laminate	Tag B Laminate
Value	0	0	21	24

TABLE I: Averaged RFID moisture values of four tags on two different surfaces.

[21]. It was found that mounting them to different surfaces changed their nominal moisture value. This can be attributed to the material the tags are mounted on changing the electrical permittivity of the tags environment and, therefore, the tag's moisture values. The result of a preliminary experiments are shown and briefly described in Table I. For this brief demonstration, average Moisture value were taken from 10 measurements in each surface from two different tags. All measurements were taken at similar temperatures and humidity and the experiment was conducted within a 5 minute interval.

Bending the tag also greatly influenced the tags reported moisture values. Indeed, Smartrac notes this phenomenon and market it as potential applications for their RFID Dogbone tags [22]. Increasing the bend in a tag can be modeled as a change in angle of a capacitors plates. The plates of the capacitor being the two sides of the tag. The greater the angle of the bend, the closer the two sides of the tag are to being parallel as shown below. This can cause an issue in the field where tags could be accidentally bent during installation or when soil shifts, resulting in an undetectable failure.

It was decided that rigid housings made from a specified material would be the best way to prevent these factors from interfering with the tags values once they are planted. Plated aluminum, ABS, PLA, and plexiglass were all considered. ABS plastic (via 3D printing) was a chosen because it is easier to make custom designs than with plexiglass (via laser cutting). A conductive enclosure would form a Faraday shield, preventing the tags from communicating with the antenna. After testing the pliability of 3D printed ABS at multiple thicknesses, a thickness of 1mm was chosen and used ubiquitously for all experiments unless otherwise stated. A 3D printing method of manufacturing was used because it allows for fast iterations and more freedom of experimentation with varying thicknesses, however, injection molding, or laser

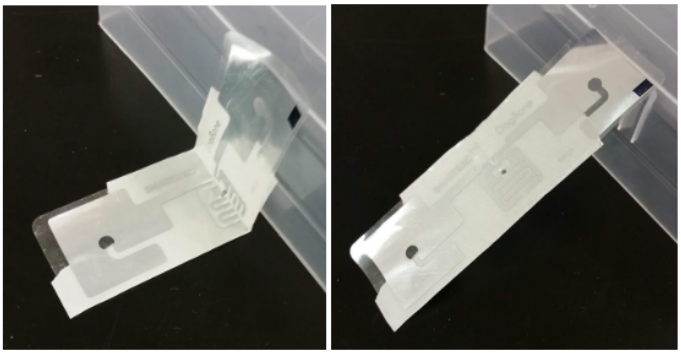

Fig. 2: The RFID tag was shown to exhibit different moisture values when flat and curved (21 and 31 respectively)

cutting or punching a sheet of known thickness would be better suited manufacturing techniques as they create parts with more consistency and fewer defects.

It is important to note that the distance from the antenna to the tag or objects between the tag and antenna did not produce any perceivable interference; although no, formalized experiments were undertaken to prove this. It was not seen as an issue during experimentation. This is also supported by their datasheet [1].

A. Housing Build-Guide

To ensure the replicability of experiments mentioned in this thesis, the exact method of tag construction used is briefly mentioned in this section. In order to construct a single tag the following resources are required:

- Access to a 3D printer with ABS plastic
- ABS Cement (ABS saturated acetone)
- One Smartrac Dogbone RFID Tag (model number 401)

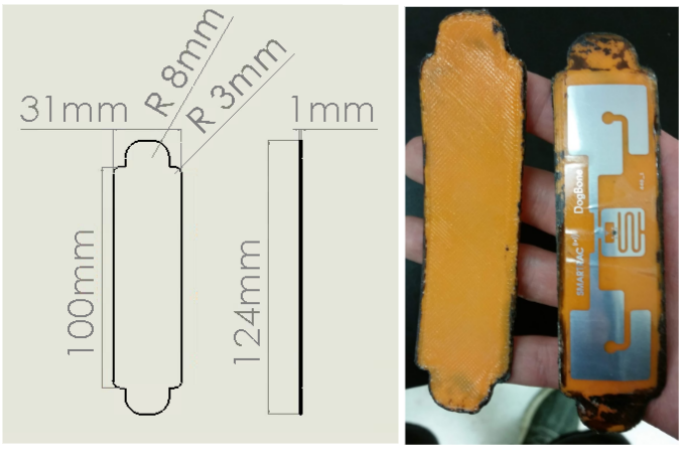

Fig. 3: LEFT: Fully defined dimensions of the RFID housing used in tested design. RIGHT: How the tag sits on one sides of the housing. The unused area around the tag and bulbs at the top and bottom provide area for the glue to better seal the sandwich.

A set of two identical paddles are printed to make a single RFID tag. The tag is attached to one side of the housing. The second side is then glued onto the first via ABS cement via the tabs and edges to complete an RFID sandwich. Care should be taken to avoid getting the glue onto the tag itself, as the solvent of the glue could erode the plastic of the tag or could create lumps in the middle of the tag. Once the tags are attached. After this design was finalized and built, a small fleet of sensors were constructed for calibration.

IV. Experimental Methods

A. Calibration Setup

Dogbone RFID tags output unitless values for moisture. Prior to this experiment, commercial tags utilizing Chameleon technology were never used to sense soil moisture. Therefore, no equation existed to easily convert their sensor readings to VWC. A significant portion of the inner workings of a tag are industry secrets–such as the exact metal used, and the pin impedance properties of the Dogbone IC. This makes modeling infeasible. Therefore, it was decided that determining the relationship between the Dogbones 5-bit sensor code and VCM would be done using empirical measurements. Measurements were taken by comparing moisture levels measured by the Dogbone to an industry standard soil permittivity sensor.

A Decagon 5TM soil moisture probe was chosen as the sensor to proceed with this calibration because of its high accuracy and availability (Decagon merged with Meter Group). The 5TM uses the frequency response of a probe submerged in the medium being measured as a way of determining the relative permittivity of the soil [25][26]. Relative permittivity is unitless metric. Its calculated as a ratio of absolute permittivity over the permittivity of a vacuum. It is a material property that affects the force between two charges in a material as described by Coulombs law. It is also sometimes called dielectric constant of a material.

Using the equation derived by Topp (et al.) shown in Figure 8, the VWC can be derived from the relational permittivity [6]. Per Decagon, their 5TM sensor works within 3% of the actual VWC in soil when properly installed and not compensated for specific soil types (Decagon 5TM manual).

$$\theta = 4.3 * 10^{-6}\varepsilon_r^3 - 5.5 * 10^{-4}\varepsilon_r^2 + 2.92 * 10^{-2}\varepsilon_r - 5.3 * 10^{-2}$$

Fig. 4: Topp equation relating dielectric permittivity (ε_r) to volumetric water content (θ)

$$\varepsilon_r = \frac{\varepsilon_{raw}}{50}$$

Fig. 5: Equation relating 5TM raw measurement to true dielectric permittivity

Both the tag and probe were buried three-to-five inches deep in a homogeneous mixture of sandy loam soil retrieved from nearby the lab. The tag was installed parallel to the surface and antenna to maximize read range with the antenna. An example of this setup is shown in Figure 6. The 5TM probe was installed horizontally into undisturbed soil, also parallel

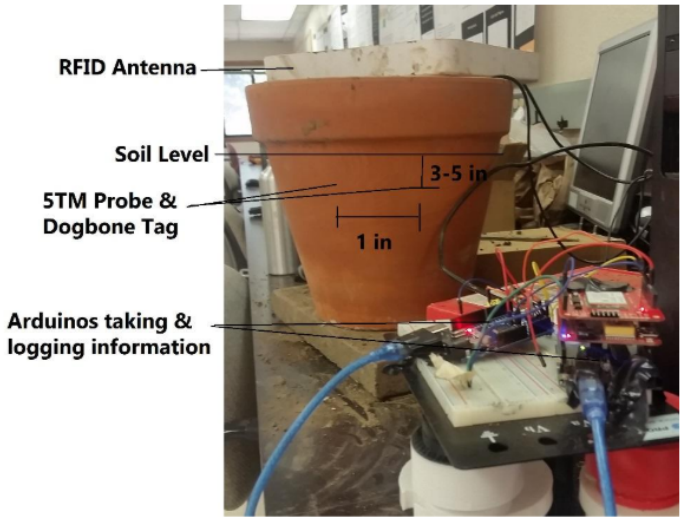

Fig. 6: The experimental setup. The 5TM and Dogbone were completely buried 7-14cm below the surface of the soil and 2.5 cm away from each other.

to the surface, to best mimic the tags position. The 5TM was installed following the installation description in the manual [25].

Data from the Dogbone and Decagon 5TM sensor were read periodically at a 20 second interval over a time of seven minutes. Soil was measured with a range of relative permittivity from 5 to 18 ε_r (VWC of 9% to 31%). This corresponded to a large enough range to derive conclusions, although a larger range would have been ideal. Measuring the RFID tags above a relative permittivity 18 was determined to be impossible with the described setup. Presumably, this was caused by the inability of the antennas signal to penetrate the soil past that threshold of soil dielectric permeability. Also, at that relative permittivity, the Dogbone tags transmitted a lowest possible value of 0 creating an artificial ceiling for all further readings with the enclosure thickness tested (1mm each side).

Moisture data from the Dogbone tags was read using a standard Class-1 Generation-2 (C1G2) read command [1]. This was accomplished using an Arduino UNO with a specialized UHF RFID shield with an embedded ThingMagic M6E-Nano RFID chip. The board was programmed using the Arduino IDE and leveraged a modified version of SparkFuns SparkFun Simultaneous RFID Tag Reader Library [27]. This setup was operated at a power setting of 26 dBm. The antenna was an ALIEN ALR-8698 RFID antenna with a gain of 11.0 dBic. It was connected using an onboard u.FL connector.

To log data, information was taken from each sensor by a set of Arduinos communicating over I2C. Two Arduinos were needed because the library needed to communicate with the 5TM probe was not compatible with the SoftwareSerial library needed for reading the RFID tags. One Arduino was set as a slave on the I2C line and read data from the 5TM sensor. The other master Arduino had both a SD logger, and a RFID reader shield on it.

B. Experimental Procedure

To conduct each trial analyzed to build the model, the following procedure was conducted to gather trials. The same soil sample was used for all trials.

1) Dry the soil by leaving it in the sun for two or more days
2) Break up any hardened clumps in the soil using a metal rod
3) Install the RFID Tag and the 5TM probe into the soil (as described in Calibration Experiment Setup).
4) Take 20 readings of both the 5TM and RFID tag with a period of 20 seconds between each measure
5) Remove the RFID Tag and 5TM probe from the soil
6) Perturb the soil with a metal rod for a full minute
7) Repeat steps 3 through 6 three to five times (this is done to help account for install errors)
8) Add between 20g and 100g of water the the soil and mix until the soil is at a homogenous VWC throughout, at least a full minute.
9) Repeat steps 3 through 8 until the RFID Tag starts to display a zero value or cannot be read.

This full procedure was conducted three times to gather the dataset analyzed in further sections. Initial tests of this system showed noticeable variance between moisture measurements taken in rapid succession for the SmarTrac Dogbone tags. This variance was mitigated by taking the average of multiple sensor readings (20) taken in quick succession. A tag reading takes 4ms and so this change was imperceptible [1]. This increase of time for each sensor reading did not have a practical impact the perceptive speed of the RFID-based system or hinder its ability to function. Future experiments should test variations and the exact effect of this practice on reading accuracy and precision.

V. Results

Using the method described, 49 trials were conducted at a VWM range of 1-31% VWM, as measured by the Decagon 5TM VWM probe. The first five measurements for each trial were collected and analyzed.

Within the dataset of the median values of trials, there appears to be three outliers that all read RFID moisture values above 25. Generally, outliers in a dataset can be attributed to four causes: errors with data entry, the dataset being incomplete, errors with sampling, and an intrinsic, extreme distribution in the variables.

Due to the nature of soil capacitance is that extreme care must be taken during installation to avoid misreadings. Although, steps were taken to prevent this as described in the Methods section, the possibility of an improper installation of either the RFID sensor or the 5TM is high. Such an improper install during a trial would affect the accuracy but not the precision of readings; the mean value would shift with the variance remaining consistent with the general trend for said trial. It is believed that these outliers represent installation errors and should be removed from the final model because they are not representative of the true case.

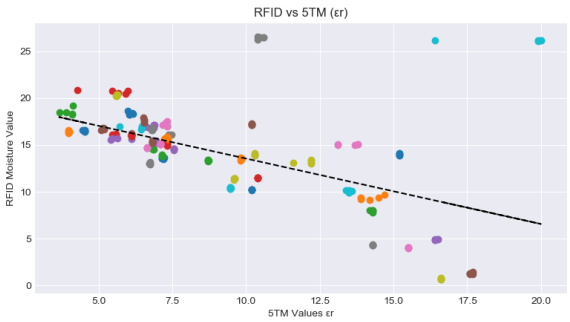

Fig. 7: RFID moisture values vs ε_r measured by a Meter 5TM soil moisture probe. Outliers were omitted when calculating the regression line.

With these three outlier trials removed from the set, we calculated a linear relationship for the RFID Moisture Value and VWC using least squares regression. This resulted in an R-squared values of 0.72 with suspected outliers removed, and an R-squared value of 0.29 with outliers included.

$$f(x) = -0.6977x + 20.52$$

Fig. 8: Function to convert RFID values to ε_r. This can be converted to VWC through the Topp equation.

A. Issues with Data Collection

It can be seen that the domain of values collected in this experiment is fairly limited: a VWC range of 1-31% omits values on either extreme. The lack of any measurements at 0% was caused by the act of drying samples in the sun, which likely left some residual water. This could be prevented by drying samples in an oven at 60-70C for at least 48 hours, as recommended by Decagon [25]. The gap in the domain can be attributed to how imprecisely water was added to the soil. For futures tests, this can be be prevented by consistently incrementing the soil moisture with a measured volume of water instead of the ad-hoc eyeball method implemented in this experiment.

The lack of any samples above 31% VWC was caused by limitations of the implemented RFID system. Above this VWC, the RFID Tags moisture value reached its minimum value at zero (truncating measurements) and the number of successful antenna reads became unusably low. Using a thicker RFID enclosure may work to decrease the tags sensitivity and prevent it from truncating low VWCs, although this may also decrease the accuracy. Reading tags reliably above 31% VWC may also require either increasing the antenna gain or the power of the RFID Tag reads. This would require swapping out hardware for more expensive components.

B. Proposed Filtering Technique

After close analysis of the data, there was determined to be an issue with the 5TM sensor where approximately 10% of

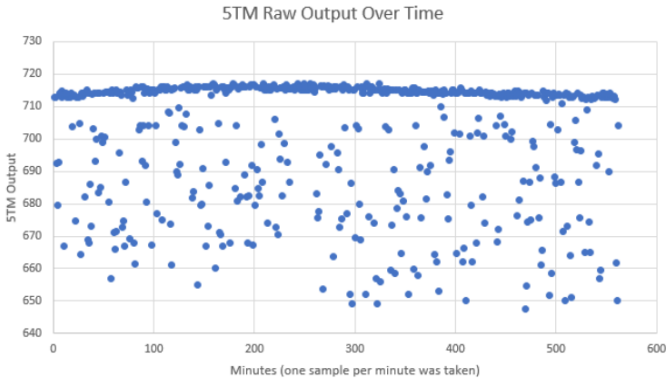

Fig. 9: Graph illustrating left-skew distribution in consecutive 5TM probe readings in an undisturbed soil sample.

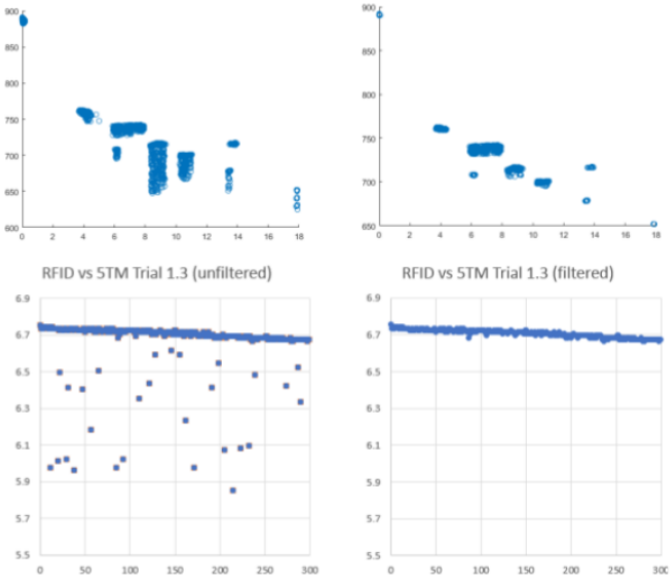

Fig. 10: Illustrations of the effect of proposed filter. Unfiltered (left) vs filtered data (right) for a sample of effected trials (above) and a single zoomed-in trial (below).

readings were skewed left. This noise is illustrated in Figure 9.

To mitigate the effect of this assumed noise, a filtering technique for 5TM data was developed. It uses a concept of a local filter or filter window that removes data beyond a set range of the local median for each trial. If the distance from the median was greater than a small buffer of 4 ε_r, then that data point was discarded. This created a Gaussian distribution around the median. The effect of this filtering can be seen in FIGURE 10 and the code for it can be found in the appendix.

VI. Field Experiment

To prove that this system works when translated to the field, the RFID reader and a 15 tags were installed on a quarter segment of a 10m center pivot arm at Oregon State Universitys Agricultural Research and Extension Center in Hermiston, Oregon.

Fig. 11: Image from the field trial, taking place on a 10 meter radius center pivot irrigation arm.

Fig. 12: (Left) Image from the field of the primary researcher collecting field data. Notice the flags used to identify where tags were planted. (Right) image from the custom app that overlaid tag information in Google Maps with a Satellite view active to show the center pivot arm.

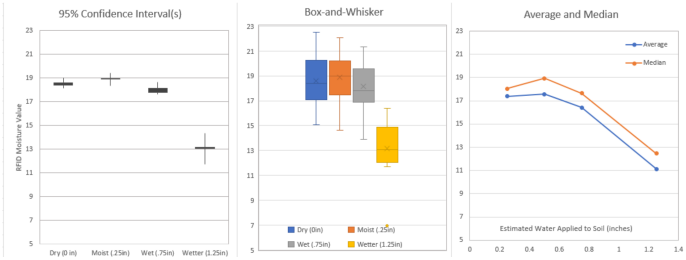

Fig. 13: Data from the field trials visualized.

These tests were inspired by the tests done in by Jason Kelly for his phd **??**. The buckets were spaced evenly across the field, the field was irrigated, and then the buckets were weighted to determine the exact quantity of water applied at that particular spot in the field. This was done to test for areas in which the field was over or under irrigated. We hoped replicate the conclusions of these tests with the RFID Tags in lieu of the buckets with the RFID Tags having the additional benefit of measuring the water in the soil rather than just the volume of water applied. This would account for soil properties which the bucket test cannot: subsurface fluid flows, the water retention of the soil, ect.

Before readings were taken, the following procedure was undertaken to plant the tags. Soil was compressed by two individuals applying 140 to 210 lbs delivered across an area of $200cm^2$ or $215cm^2$ respectively via a gentle press of 2 second with tennis shoes (US size 10 or 12.5). The tags were then planted 4 inches ($10cm$) deep in this compressed soil, using ruler for reference. 15 tags were distributed around $1/4$ of a turn around the arm as shown in FIGURE 12. The locations of each named tag were recorded and labeled with a small flag so they could be located.

Initial samples were collected by a handheld version of the RFID Reader which was made by powering the system with a portable phone charger with a maximum current supply of 1 amp. The center pivot was programmed to deliver $1/4$ to $1/2$ inch of water. It took about 45 minutes for the arm to complete this pass. Data from every tag was collected 20 minutes after the pass completed to allow the water time to fully permeate. The tags values were then collected after waiting 10 minutes to allow the applied water to permeate. The application of water via the center pivot, waiting 30 minutes, and then recording the RFID Tag values was conducted three times. The final dataset contained RFID values after the center pivot had applied 0, 0.25, 0.75, and 1.25 inches of total water.

Upon analysis of the samples, the tags showed a strong correlation between the amount of water applied and the RFID Moisture Value. Although this trend averaged over all of the tags was strong, there were significant inconsistencies between different tags. These inconsistencies could have been caused by any of the following known variables: inconsistent application of water by the center pivot arm, inhomogeneous soil pack, issues with building the RFID Tags housing, evaporation, or variations of temperature over the course of the experiment.

It appears that at 1.25 inches of applied irrigation, water perimated down and affected tag measurements. This was statistically examined by calculating p-values for each irrigation application compared to the result after the previous application using a paired sample, two-tail t-test. A table of these p-values is shown in Table II. This supports the rejection of the null hypothesis in favor of the alternative for the applied irrigation step between 0.75in and 1.25in. From a frequentist standpoint, this suggests that the wetting front reached the tags during this step.

VII. Conclusion

The result of calibration experiments suggest a highly correlated relationship between the Dogbone RFID tags and the electrical permittivity of the soil exists. The existence of such a relationship supports the original hypothesis that the Dogbone RFID tags can be used as a soil moisture sensor. There was shown to be a statistically significant trend of an R^2 value of 0.7 with outliers omitted.

However, the existence of outliers makes tag readings difficult to determine the exact electrical permittivity from the Dogbone moisture value for any single measurement. The inhomogeneity of soil as a medium can lead to serious errors if care is not taken to install sensors properly and consistently. It should be noted that this is a problem for all capacitive soil

Irrigation (in)	0-¼	¼-¾	¾-1 ½
T-test Probability	38%	1.1%	0.015%

TABLE II: Two tailed, paired t-test probability examining changing RFID moisture values in relation to changes in total applied irrigation amounts.

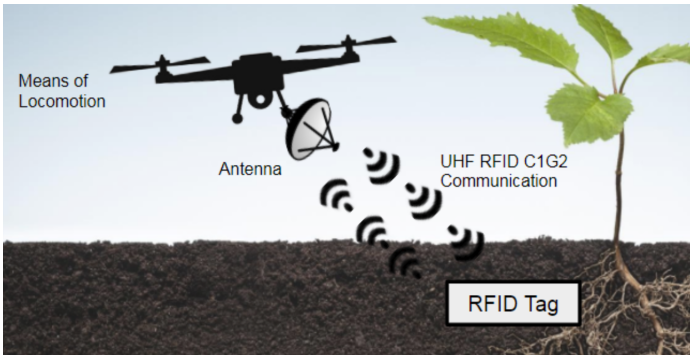

Fig. 14: The proposed system consisting of a means of locomotion, an antenna, and an RFID Tag. A drone is shown, but it could just as well be a tractor, center pivot or linear irrigator, or a low flying aircraft.

moisture sensors [25]. Still, it presents a sizable hurdle for attempting to integrate or utilize these sensors as a critical piece of a larger system.

A. Future Experiments

Due to the high level of promise that the Dogbone RFID Tags show for use in environmental sensing applications, it is suggested that future experiments be conducted to more fully evaluate the RFID Tags potential. Tests involving more data point gathered at lower VWCs would provide a better idea of the tags performance in desiccant environments. This could be achieved by using an oven to dry soil samples to completion rather than using the sun which may have left residual moisture intact. All calibration experimentation here was done on a single soil sample of unknown composition. In comment on the usability of tags in all soil types, they should be further evaluated in soil samples of known composition.

It is suggested that future testing on the tags should use laboratory methods of measuring VWC from gravimetric water content (GWC). Such an experiment would involve weighing soil samples after an RFID tag value is recorded, drying the samples in an oven at 60-70C for 48 hours, and then re weighing the sample. Using the equation $GWC = {}^{WET-DRY}/_{DRY}$, GWC can be calculated . From that, the bulk density (BD) of the soil must be calculated by dividing the dry weight by the volume of the sample $BW = {}^{DRY}/_{VOL}$. Finally, VWC can be found by multiplying GWC and BW together $VWC = GWC * BW$.

An interesting use case for the RFID Tags is to detect moisture content of snow. This could be used to improve the accuracy of LIDAR measurements on snowpack. In order to be used in this way, the tags would need to be tested and verified as the low temperature of snow may mess with the tags electronics making them inoperable. However, other RFID tags have been shown to work at low temperatures [28].

As noted, the tests done in this thesis made use of soil samples with unknown physical properties. This did not control for confounding variables such as salinity, particulate size, and presence of macro- and micro-pores. In future experiments, validation or calibration tests could be conducted in accusands instead of in random sample of soil. The use of accusands is advantageous because it has known consistent physical properties which make it an excellent media for soil experiments [29].

Another alternative to calibrating soil moisture sensors directly in soil is to calibrate in a liquid of known relative permittivity. The procedure would be similar to procedures done by Bogena (et al), and Robinson (et al) [30][5]. Potential liquids to be used include isopropoxyethanol, dioxane, limonene, and deionized water. These chemicals have well-tested and understood relative permittivities. A full range of relative permittivities that correspond to the entire VWC range (ε_r from 3-80) could be constructed [5]. By diluting dioxane and isopropoxyethanol with deionized water properly, any ε_r could be achieved. One issue with this approach is that the chemicals may dissolve the tag housings; this could be mitigated by using a resistant plastic, like PTFE or HDPE, and minimizing time spend in the media by taking measurements quickly. As a while, using liquids over soil or physical substrate for calibration has several key advantages that include a lack of air gaps, almost guaranteed homogeneity, and ease of preparing a sample.

B. Potential Improvements

There are several major improvements that could be made to this system given more time and resources. Several especially feasible improvements stand out as ways to significantly improving the system.

The most constraining limitation of current system is the maximum depth at which RFID moisture sensors can take readings. This is primarily caused by electromagnetic communication signals being absorbed by soil, preventing the tag from receiving enough power to be read. Increasing the signal strength will allow for an increase in the maximum reading depth, but will cause increase the cost and power needs of the reader.

Incorporating a monopole onto the tag would greatly increase the usability of this system. This has already been successfully implemented and proven on tags without the self-tuning capability [16][18]. Having such a feature in conjunction with a self-tuning IC would allow for the reading of soil moisture at a specific depth or range of depths at a greater depth than the current system is limited to. It would also take the average relative permittivity over a larger area which would potentially reduce errors caused by soil inhomogeneity.

Encasing the moisture sensitive area of the tag in gypsum (or similar moisture-absorbing medium) is another potential improvement. This would read the water tension in the soil rather than VWC. Water tension is considered by some to be more important to plant growth than VWC as it is an indication of how much water is available to the plant rather than the total amount of water in the soil (which may be present in inaccessible micro-pores) [31].

The RFID housing could be improved by using sheet plastic insead of 3D print to avoid small defects intrinsic with 3D printing. Could be stamped or laser cut. This would cost similarly to 3D printing. Injection molding would be the best

Fig. 15: Primary author, Brett (right), working alongside Jonah Siekmann (left) to attach the RFID reader antenna onto a octocopter drone– an effort that turned out to be only semi-successful due to a design flaw in the drone and RFID read range limitations.

method for large scale production, but requires significant upfront costs.

Working out a method for consistently planting tags at depth and angle should be explored. Its theorized that this could be automated by a robotic arm, like the FarmBot CNC machine [32].

Developing means of harvesting the data from these sensors will be vital for the creation of a marketable system. There are several possible methods for creating such a harvesting device, the most feasible being mounting the RFID reader onto a piece of irrigation equipment, such as a center pivot or wheel move system. Attaching the RFID reader onto a another means of locomotion such as a tractor, drone, ATV, or custom robot also holds a high potential.

In parallel with this effort, Jonah Sieckmann, a fellow OPEnS researcher, designed an octocopter drone to carry the reader. Multiple preliminary and a final BOM can be found at the google doc cited here and also in the appendix [33]. The chosen final version was not able to carry the reader system stability. It broke after a crash on the first test flight [34]. Jonah believes that this was caused by the battery voltage being too low for the chosen motors (too much mAh and too little V) and the motors being of poor quality (kinda sucked).

REFERENCES

[1] *SENSOR DOGBONE Moisture Level Sensing Inlay*, Smartrac Group, May 2018, retrieved from https://www.smartrac-group.com/files/content/Products_Solutions/PDF/0027b_SMARTRAC_SENSOR_DOGBONE.pdf.

[2] C. A. Dieter, M. A. Maupin, R. R. Caldwell, M. A. Harris, T. I. Ivahnenko, J. K. Lovelace, N. L. Barber, and K. S. Linsey, "Estimated use of water in the united states in 2015," 2018. [Online]. Available: https://doi.org/10.3133/cir1441

[3] United States Department of Agriculture, "Irrigation water use," Tech. Rep., April 2019.

[4] F. S. Zazueta and J. Xin, "Soil moisture sensors," *Soil Science*, vol. 73, pp. 391–401, 1994.

[5] D. Robinson, C. Campbell, J. Hopmans, B. K. Hornbuckle, S. B. Jones, R. Knight, F. Ogden, J. Selker, and O. Wendroth, "Soil moisture measurement for ecological and hydrological watershed-scale observatories: A review," *Vadose Zone Journal*, vol. 7, no. 1, pp. 358–389, 2008.

[6] G. C. Topp, J. Davis, and A. P. Annan, "Electromagnetic determination of soil water content: Measurements in coaxial transmission lines," *Water resources research*, vol. 16, no. 3, pp. 574–582, 1980.

[7] D. Entekhabi, E. Njoku, P. O'Neill, K. Kellogg, and J. Entin, "The nasa soil moisture active passive (smap) mission formulation," in *2011 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2011, pp. 2302–2305.

[8] Y. H. Kerr, P. Waldteufel, J.-P. Wigneron, J. Font, and M. Berger, "The soil moisture and ocean salinity mission," in *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No. 03CH37477)*, vol. 1. IEEE, 2003, pp. 1–3.

[9] N. JPL. (2015) Nasa soil moisture radar ends operations, mission science continues. [Online]. Available: https://www.theguardian.com/world/2017/mar/12/netherlands-will-pay-the-price-for-blocking-turkish-visit-erdogan

[10] C. A. Walton, Patent.

[11] J. Yin, J. Yi, M. K. Law, Y. Ling, M. C. Lee, K. P. Ng, B. Gao, H. C. Luong, A. Bermak, M. Chan *et al.*, "A system-on-chip epc gen-2 passive uhf rfid tag with embedded temperature sensor," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 11, pp. 2404–2420, 2010.

[12] E. M. Amin, J. K. Saha, and N. C. Karmakar, "Smart sensing materials for low-cost chipless rfid sensor," *IEEE Sensors Journal*, vol. 14, no. 7, pp. 2198–2207, 2014.

[13] P. Gou, N. D. Kraut, I. M. Feigel, H. Bai, G. J. Morgan, Y. Chen, Y. Tang, K. Bocan, J. Stachel, L. Berger *et al.*, "Carbon nanotube chemiresistor for wireless ph sensing," *Scientific reports*, vol. 4, p. 4468, 2014.

[14] L. Kriara, M. Alsup, G. Corbellini, M. Trotter, J. D. Griffin, and S. Mangold, "Rfid shakables: Pairing radio-frequency identification tags with the help of gesture recognition," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 327–332.

[15] C. Bauer-Reich, J. Hoey, R. Sailer, N. Schneck, and C. Ulven, "Biodegradable soil sensor, system and method," Patent.

[16] A. Hasan, R. Bhattacharyya, and S. Sarma, "Towards pervasive soil moisture sensing using rfid tag antenna-based sensors," in *2015 IEEE International Conference on RFID Technology and Applications (RFID-TA)*. IEEE, 2015, pp. 165–170.

[17] T. K. Hamrita and E. C. Hoffacker, "Development of a smart wireless soil monitoring sensor prototype using rfid technology," *Applied Engineering in Agriculture*, vol. 21, no. 1, pp. 139–143, 2005.

[18] N. S. da Fonseca, R. C. Freire, A. Batista, G. Fontgalland, and S. Tedjini, "A passive capacitive soil moisture and environment temperature uhf rfid based sensor for low cost agricultural applications," in *2017 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*. IEEE, 2017, pp. 1–4.

[19] S. Pichorim, N. Gomes, and J. Batchelor, "Two solutions of soil moisture sensing with rfid for landslide monitoring," *Sensors*, vol. 18, no. 2, p. 452, 2018.

[20] Q. Wan, R. K. Kanth, G. Yang, Q. Chen, and L.-R. Zheng, "Environmental impacts analysis for inkjet printed paper-based bio-patch," *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, vol. 2, no. 4, 2015.

[21] C. Swedberg. (2014, April) Smartrac group and rfmicron to develop passive sensor tags.

[22] *AN-FAM1601, Passive Sensors Technical Guide, Application Note*, Smartrac Group, May 2017, retrieved from https://www.smartrac-group.com/files/content/Products_Solutions/PDF/Passive%20RFID%20Sensors%20Technical%20Guide_AN-FAM-1601_web.pdf.

[23] D. R. Lide, *CRC handbook of chemistry and physics*. CRC press, 2004, vol. 85.

[24] C. Swedberg. (2014, December) Advancement through technology: Audi cars carrying smartrac tags.

[25] *5TM Manual Web*, Meter Group, 2010.

[26] U. Rosenbaum, J. Huisman, A. Weuthen, H. Vereecken, and H. Bogena, "Sensor-to-Sensor Variability of the ECHO EC-5, TE, and 5TE Sensors in Dielectric Liquids," *Vadose Zone Journal*, vol. 9, p. 181, 2010.

[27] B. Stoddard, "Sparkfun simultaneous rfid tag reader library," https://github.com/stoddabr/SparkFun_Simultaneous_RFID_Tag_Reader_Library, 2018.

[28] J. Nummela, L. Ukkonen, and L. Sydanheimo, "The effect of low temperature on passive UHF RFID tags," in *Proceedings of the 4th WSEAS International Conference on REMOTE SENSING (REMOTE'08)*, 2008, pp. 88–92. [Online]. Available: \url{https://www.researchgate.net/publication/265074862_The_Effect_of_Low_Temperature_on_Passive_UHF_RFID_Tags}

[29] M. Schroth, J. Istok, S. Ahearn, and J. Selker, "Characterization of miller-similar silica sands for laboratory hydrologic studies," *Soil Science Society of America Journal*, vol. 60, no. 5, pp. 1331–1339, 1996.

[30] H. Bogena, J. Huisman, B. Schilling, A. Weuthen, and H. Vereecken, "Effective calibration of low-cost soil water content sensors," *Sensors*, vol. 17, no. 1, p. 208, 2017.

[31] E. A. Czyż and A. R. Dexter, "Plant wilting can be caused either by the plant or by the soil," *Soil Research*, vol. 50, no. 8, pp. 708–713, 2013.

[32] J. Cruz, S. Herrington, and B. Rodriguez, "Farmbot," 2014.

[33] "Drone bom," https://docs.google.com/spreadsheets/d/1eLcoSJWrwk96wnhvx_0th_iSSQr9AySgDCfaHJUDVm0/edit?usp=sharing.

[34] "Hydrone slideshow (public)," https://youtu.be/DyCMxGsqV98?t=153.

Appendix

Appendix Table of Contents

Commercialization Plan

With the inexpensive and commercially available setup we put together, the Dogbone RFID Tags are capable of reading soil moisture reliability at a depth of 4 inches for the VWC interval tested. When planted deeper, the RFID Tags were not consistently readable at high levels of VWC. According to Green, soil moisture is most important up to a depth of 30 inches [Green]. However, the early stages of plant growth--emergence, flowering, and formation--are generally more sensitive to soil moisture than vegetative growth after establishment--early and late growth periods [Johl]. Therefore, it is important to track the moisture content of the soil at the root depth of these early growing periods. This gives the RFID Tags a use case for ensuring that soil contains the right amount of moisture during early growth stages.

Due to the practicality of this method, serious thought was put into what the commercialization of this system would entail. A system comprising of three main components was derived, as shown in FIGURE 14. In this system, the reader is mounted to a means of locomotion (drone, tractor, rail, by hand for instance) where it transmits a signal and reads tags that are within range. The antenna's signal feeds the tag with enough energy to briefly power on. The signal also interacts with the surrounding soil which can be measured to produce a moisture value. The tags then modulate the reader's signal to communicate the this sensor value along with the tag's identification number which is mapped to a known GPS location.

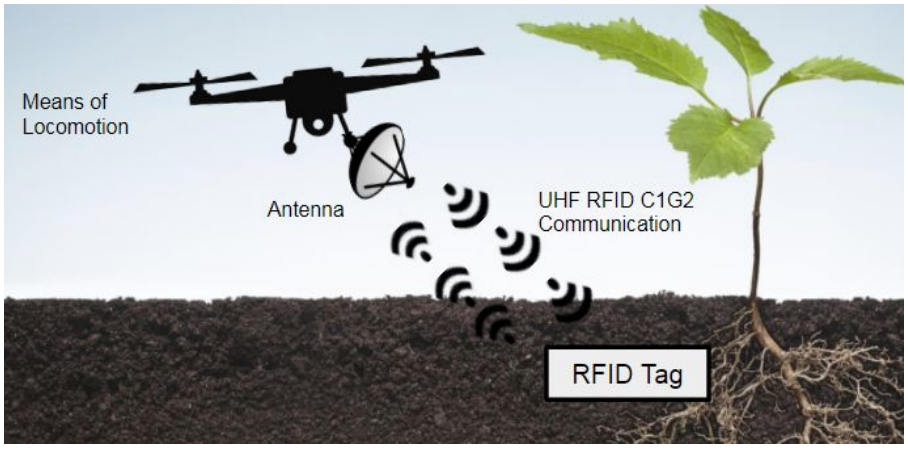

Fig. The proposed system consisting of a means of locomotion, an antenna, and an RFID Tag. A drone is shown, but it could just as well be a tractor, center pivot or linear irrigator, or a low flying aircraft.

Developing this system into a minimum viable product (MVP) involves three main costs: people, equipment, and travel. Assuming there are four researchers working 10 hours average a week at an average wage of $15/hr it will cost $20k for eight months of employment, with another $20k in other employment costs to the company such as health benefits, and HR costs. The estimated cost of equipment includes $3k for a heavy lift drone, $10k for various RFID reader

modules and other hardware, and $2k in softwares costs. Sending two researchers to the Hermiston test site five times over the course of the field test will cost a total of $5k assuming it costs $500 per researcher. This brings the total price to develop an MVP to $60k.

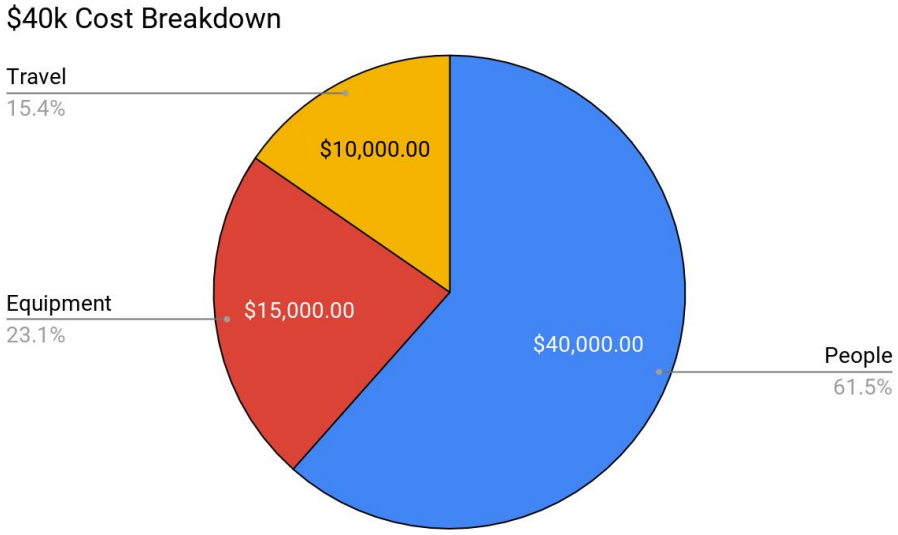

Fig. Cost breakdown of developing this system into a full commercial product.

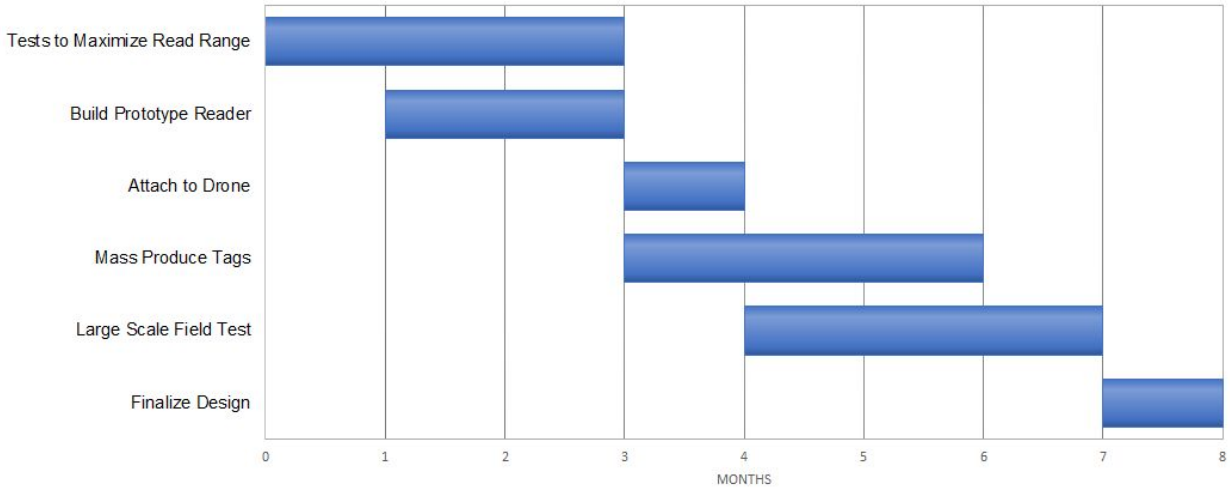

Fig. Gantt chart for commercializing the sensors given $40k in grant funding.

Drone BOM

		Final Build:			
CC3D FC	$19	1	$19	Amazon	Flight Controller
750kv Turnigy Motors	$16	4	$64	HobbyKing	High-Torque, efficient motor.
30A Esc (4pck)	$25	1	$25	Amazon	Electronic Speed Controller
500mm Frame	$34	1	$34	Amazon	Frame
5000mah 4s batt	$33	1	$34	Hobby King	Battery
12x4.5 props	$8	1	$8	Amazon	Propellors
		Total:	$184		
Arduino Uno	$11	1	$11	Amazon	Mission Controller
GPS module	$24	1	$24	Amazon	GPS
RC Transmitter/receiver	$53	1	$53	Amazon	Killswitch
Sonar (altitude)	(I have a few laying around that cost <$2 that we can use)				
Barometer	$10	1	$10	Adafruit	
Magnetometer	$10	1	$10	Amazon	
		Total:	**$108**		

Table of Calibration Values

Test ID is a combination of the day the trial was conducted, the level of wetness, and the trial number of that wetness. For example, the second trial done at the dryest level of soil moisture done on August 10th will be “8-10 2LT1”.

Test ID	RFID	5TM	RFID VWC	5TM VWC
8-10 1LT1	16.4	4.54	0.169963	0.068634
8-10 1LT1	16.55	4.46	0.167836	0.066673
8-10 1LT1	16.45	4.54	0.169255	0.068634
8-10 1LT1	16.55	4.54	0.167836	0.068634
8-10 1LT1	16.6	4.53	0.167125	0.068389
8-10 1LT2	16.3	3.97	0.171376	0.054525
8-10 1LT2	16.45	3.99	0.169255	0.055025
8-10 1LT2	16.45	3.97	0.169255	0.054525
8-10 1LT2	16.25	3.97	0.172081	0.054525
8-10 1LT2	16.6	3.97	0.167125	0.054525
8-10 1LT3	18.25	4.11	0.14306	0.05802
8-10 1LT3	18.45	3.89	0.140062	0.052518
8-10 1LT3	18.45	3.66	0.140062	0.046715
8-10 1LT3	18.3	4.11	0.142312	0.05802
8-10 1LT3	19.2	4.12	0.128666	0.058269
8-10 2LT1	20.75	6	0.104313	0.103329
8-10 2LT1	20.85	4.28	0.102705	0.062238
8-10 2LT1	20.55	5.65	0.107517	0.095198
8-10 2LT1	20.8	5.47	0.10351	0.090971
8-10 2LT1	20.45	5.9	0.109112	0.101018
8-10 2LT2	17.1	6.87	0.159956	0.12304
8-10 2LT2	16.85	6.86	0.163554	0.122817
8-10 2LT2	17.1	6.88	0.159956	0.123262
8-10 2LT2	16.9	6.65	0.162837	0.118122
8-10 2LT2	17.15	6.89	0.159233	0.123485

8-10 2LT3	16.65	5.19	0.166413	0.084334
8-10 2LT3	16.7	5.19	0.1657	0.084334
8-10 2LT3	16.6	5.07	0.167125	0.081467
8-10 2LT3	16.85	5.16	0.163554	0.083619
8-10 2LT3	16.85	5.16	0.163554	0.083619
8-10 3LT1	17.05	7.32	0.160678	0.13296
8-10 3LT1	17	7.32	0.161399	0.13296
8-10 3LT1	17.1	7.16	0.159956	0.129454
8-10 3LT1	17.55	7.31	0.153413	0.132742
8-10 3LT1	17.4	7.28	0.155603	0.132086
8-15 1L1T	16.1	7.48	0.174189	0.136443
8-15 1L1T	16.1	6.12	0.174189	0.10609
8-15 1L1T	16.25	6.11	0.172081	0.10586
8-15 1L1T	16.05	6.1	0.174889	0.105631
8-15 1L1T	16.1	7.36	0.174189	0.133833
8-15 1L2T	20.2	5.59	0.113079	0.093793
8-15 1L2T	20.2	5.62	0.113079	0.094496
8-15 1L2T	20.35	5.62	0.110702	0.094496
8-15 1L2T	20.35	5.63	0.110702	0.09473
8-15 1L2T	20.4	5.64	0.109908	0.094964
8-15 1L3T	17	6.46	0.161399	0.113839
8-15 1L3T	16.9	6.46	0.162837	0.113839
8-15 1L3T	17	6.47	0.161399	0.114065
8-15 1L3T	16.65	6.45	0.166413	0.113612
8-15 1L3T	16.95	5.7	0.162118	0.096367
8-15 1L4T	18.6	5.98	0.137803	0.102867
8-15 1L4T	18.3	6.17	0.142312	0.107236
8-15 1L4T	18.4	6.15	0.140813	0.106778
8-15 1L4T	18.2	6.04	0.143806	0.104251
8-15 1L4T	18.2	6.03	0.143806	0.10402
8-15 2L1T	15.7	7.24	0.179764	0.13121

8-15 2L1T	15.5	7.28	0.182527	0.132086
8-15 2L1T	15.65	7.22	0.180456	0.130772
8-15 2L1T	15.45	7.26	0.183215	0.131648
8-15 2L1T	15.65	7.25	0.180456	0.131429
8-15 2L2T	14.5	6.88	0.196093	0.123262
8-15 2L2T	14.6	6.83	0.194755	0.122149
8-15 2L2T	14.65	6.83	0.194084	0.122149
8-15 2L2T	14.5	6.85	0.196093	0.122595
8-15 2L2T	14.75	6.85	0.19274	0.122595
8-15 2L3T	15.2	7.32	0.18664	0.13296
8-15 2L3T	15.05	7.32	0.188682	0.13296
8-15 2L3T	15.1	7.32	0.188002	0.13296
8-15 2L3T	14.9	7.31	0.190716	0.132742
8-15 2L3T	15.25	7.31	0.185957	0.132742
8-15 2L4T	15.35	6.92	0.184588	0.124151
8-15 2L4T	15.35	6.92	0.184588	0.124151
8-15 2L4T	15.35	6.94	0.184588	0.124595
8-15 2L4T	15.45	6.94	0.183215	0.124595
8-15 2L4T	15.2	6.94	0.18664	0.124595
8-15 3L1T	17.2	10.2	0.15851	0.192181
8-15 3L1T	17.3	10.2	0.157059	0.192181
8-15 3L1T	17.2	10.2	0.15851	0.192181
8-15 3L1T	17.15	10.2	0.159233	0.192181
8-15 3L1T	17.15	10.2	0.159233	0.192181
8-15 3L2T	14.65	6.63	0.194084	0.117673
8-15 3L2T	14.7	6.65	0.193412	0.118122
8-15 3L2T	14.75	6.64	0.19274	0.117898
8-15 3L2T	14.65	6.64	0.194084	0.117898
8-15 3L2T	14.65	6.64	0.194084	0.117898
8-15 3L3T	12.9	6.74	0.216961	0.120139
8-15 3L3T	13.15	6.74	0.213767	0.120139

8-15 3L3T	12.9	6.74	0.216961	0.120139
8-15 3L3T	13	6.73	0.215687	0.119916
8-15 3L3T	13	6.73	0.215687	0.119916
8-15 3L4T	13.85	10.3	0.204694	0.194109
8-15 3L4T	13.95	10.3	0.203382	0.194109
8-15 3L4T	13.95	10.3	0.203382	0.194109
8-15 3L4T	14.1	10.3	0.201406	0.194109
8-15 3L4T	13.85	10.3	0.204694	0.194109
8-15 4L1T	10.1	13.6	0.251095	0.253208
8-15 4L1T	10.2	13.4	0.249926	0.249868
8-15 4L1T	9.95	13.5	0.252841	0.251542
8-15 4L1T	10.15	13.5	0.250511	0.251542
8-15 4L1T	9.95	13.5	0.252841	0.251542
8-15 4L2T	14.05	15.2	0.202065	0.278869
8-15 4L2T	13.9	15.2	0.204038	0.278869
8-15 4L2T	14.1	15.2	0.201406	0.278869
8-15 4L2T	14.05	15.2	0.202065	0.278869
8-15 4L2T	13.9	15.2	0.204038	0.278869
8-15 4L3T	9.35	13.9	0.259742	0.258163
8-15 4L3T	9.2	13.9	0.261447	0.258163
8-15 4L3T	9.15	14.2	0.262014	0.26305
8-15 4L3T	9.65	14.7	0.256308	0.27105
8-15 4L3T	9.35	14.5	0.259742	0.267872
8-15 4L4T	8	14.3	0.2748	0.264665
8-15 4L4T	8	14.3	0.2748	0.264665
8-15 4L4T	7.95	14.3	0.275345	0.264665
8-15 4L4T	8.05	14.2	0.274254	0.26305
8-15 4L4T	7.8	14.3	0.276976	0.264665
8-22 1L1T	16.1	5.47	0.174189	0.090971
8-22 1L1T	16	5.58	0.175589	0.093558
8-22 1L1T	16	5.58	0.175589	0.093558

8-22 1L1T	15.9667	5.58	0.176054	0.093558
8-22 1L1T	16.1667	5.58	0.173252	0.093558
8-22 1L2T	15.6333	6.11	0.180687	0.10586
8-22 1L2T	15.6333	5.43	0.180687	0.090028
8-22 1L2T	15.7333	5.47	0.179302	0.090971
8-22 1L2T	15.5667	5.42	0.181607	0.089792
8-22 1L2T	15.7333	5.64	0.179302	0.094964
8-22 1L3T	17.6	6.54	0.15268	0.115646
8-22 1L3T	17.3333	6.54	0.156575	0.115646
8-22 1L3T	17.4667	6.54	0.15463	0.115646
8-22 1L3T	17.6667	6.53	0.151701	0.115421
8-22 1L3T	17.9333	6.52	0.14777	0.115195
8-22 2L1T	15.0333	13.8	0.188909	0.256519
8-22 2L1T	15.0333	13.8	0.188909	0.256519
8-22 2L1T	15.1	13.1	0.188002	0.244801
8-22 2L1T	14.9667	13.7	0.189812	0.254867
8-22 2L1T	15	13.1	0.189361	0.244801
8-22 2L2T	26.5667	10.4	0.002813	0.196029
8-22 2L2T	26.4667	10.6	0.004699	0.199843
8-22 2L2T	26.5	10.5	0.004071	0.19794
8-22 2L2T	26.4667	10.6	0.004699	0.199843
8-22 2L2T	26.2	10.4	0.009704	0.196029
8-22 2L3T	13.4	12.2	0.210549	0.229186
8-22 2L3T	13.1	11.6	0.214408	0.218424
8-22 2L3T	13	12.2	0.215687	0.229186
8-22 2L3T	13.1333	12.2	0.213982	0.229186
8-22 2L3T	13.3	12.2	0.211839	0.229186
8-22 3L1T	26.1667	16.4	0.010326	0.296919
8-22 3L1T	26.1333	19.9	0.01095	0.344161
8-22 3L1T	26.0667	19.9	0.012192	0.344161
8-22 3L1T	26.1667	20	0.010326	0.3454

8-22 3L1T	26.1667	20	0.010326	0.3454
8-25 1L1T	13.5667	7.15	0.208389	0.129234
8-25 1L1T	13.5667	7.15	0.208389	0.129234
8-25 1L1T	13.6667	7.22	0.207089	0.130772
8-25 1L1T	13.5667	7.14	0.208389	0.129014
8-25 1L1T	13.4667	7.2	0.209686	0.130333
8-25 1L2T	15.9	7.33	0.176985	0.133179
8-25 1L2T	15.7	7.34	0.179764	0.133397
8-25 1L2T	15.8333	7.32	0.177913	0.13296
8-25 1L2T	15.7667	7.32	0.178839	0.13296
8-25 1L2T	15.6333	7.35	0.180687	0.133615
8-25 1L3T	13.8333	7.15	0.204912	0.129234
8-25 1L3T	13.8333	7.15	0.204912	0.129234
8-25 1L3T	13.8	7.15	0.205348	0.129234
8-25 1L3T	13.8333	7.16	0.204912	0.129454
8-25 1L3T	13.9333	7.15	0.203601	0.129234
8-25 1L4T	16.1	6.12	0.174189	0.10609
8-25 1L4T	15.9667	6.12	0.176054	0.10609
8-25 1L4T	16.1333	6.12	0.173721	0.10609
8-25 1L4T	16.0667	6.12	0.174655	0.10609
8-25 1L4T	15.9667	6.12	0.176054	0.10609
8-25 1L5T	14.5667	7.55	0.195201	0.137959
8-25 1L5T	14.4667	7.55	0.196538	0.137959
8-25 1L5T	14.5333	7.56	0.195648	0.138175
8-25 1L5T	14.6	7.55	0.194755	0.137959
8-25 1L5T	14.5667	7.55	0.195201	0.137959
8-25 1L6T	15.3333	6.83	0.184817	0.122149
8-25 1L6T	15.3333	6.82	0.184817	0.121926
8-25 1L6T	15.2333	6.83	0.186185	0.122149
8-25 1L6T	15.3	6.82	0.185273	0.121926
8-25 1L6T	15.4333	6.83	0.183444	0.122149

8-25 1L7T	15.0333	7.09	0.188909	0.127913
8-25 1L7T	15.1	7.09	0.188002	0.127913
8-25 1L7T	15.1	7.1	0.188002	0.128134
8-25 1L7T	15.1333	7.09	0.187549	0.127913
8-25 1L7T	15.0333	7.1	0.188909	0.128134
8-25 1L8T	16.6	6.79	0.167125	0.121257
8-25 1L8T	16.8333	6.81	0.163793	0.121703
8-25 1L8T	16.7	6.82	0.1657	0.121926
8-25 1L8T	16.6	6.83	0.167125	0.122149
8-25 1L8T	16.8333	6.79	0.163793	0.121257
8-25 2L1T	11.4667	9.61	0.234805	0.180635
8-25 2L1T	11.3	9.58	0.236829	0.18004
8-25 2L1T	11.4	9.6	0.235616	0.180436
8-25 2L1T	11.4333	9.58	0.235211	0.18004
8-25 2L1T	11.4	9.6	0.235616	0.180436
8-25 2L2T	10.3333	9.46	0.248363	0.177652
8-25 2L2T	10.5	9.47	0.246399	0.177851
8-25 2L2T	10.4333	9.49	0.247186	0.17825
8-25 2L2T	10.3333	9.47	0.248363	0.177851
8-25 2L2T	10.4667	9.47	0.246792	0.177851
8-25 2L3T	10.2333	10.2	0.249536	0.192181
8-25 2L3T	10.2333	10.2	0.249536	0.192181
8-25 2L3T	10.2333	10.2	0.249536	0.192181
8-25 2L3T	10.2667	10.2	0.249145	0.192181
8-25 2L3T	10.1667	10.2	0.250316	0.192181
8-25 2L4T	13.6	9.83	0.207957	0.184975
8-25 2L4T	13.3667	9.81	0.210979	0.184582
8-25 2L4T	13.6333	9.82	0.207523	0.184778
8-25 2L4T	13.5333	9.85	0.208823	0.185367
8-25 2L4T	13.5	9.83	0.209255	0.184975
8-25 2L5T	13.4	8.7	0.210549	0.162242

8-25 2L5T	13.3333	8.73	0.21141	0.16286
8-25 2L5T	13.3667	8.7	0.210979	0.162242
8-25 2L5T	13.4	8.7	0.210549	0.162242
8-25 2L5T	13.2667	8.71	0.212268	0.162448
8-25 2L6T	11.5	10.4	0.234399	0.196029
8-25 2L6T	11.5333	10.4	0.233993	0.196029
8-25 2L6T	11.5	10.4	0.234399	0.196029
8-25 2L6T	11.5333	10.4	0.233993	0.196029
8-25 2L6T	11.4667	10.4	0.234805	0.196029
8-25 3L1T	4.9333	16.4	0.306684	0.296919
8-25 3L1T	4.8667	16.4	0.307342	0.296919
8-25 3L1T	4.8333	16.4	0.307672	0.296919
8-25 3L1T	4.9667	16.5	0.306354	0.298379
8-25 3L1T	4.9667	16.5	0.306354	0.298379
8-25 3L2T	1.2333	17.6	0.341158	0.313995
8-25 3L2T	1.3	17.6	0.340573	0.313995
8-25 3L2T	1.2	17.7	0.341449	0.315375
8-25 3L2T	1.3333	17.6	0.34028	0.313995
8-25 3L2T	1.4333	17.7	0.339399	0.315375
8-25 3L3T	4.0333	15.5	0.315457	0.283475
8-25 3L3T	4.1	15.5	0.314816	0.283475
8-25 3L3T	4.1	15.5	0.314816	0.283475
8-25 3L3T	4	15.5	0.315777	0.283475
8-25 3L3T	4	15.5	0.315777	0.283475
8-25 3L4T	4.3667	14.3	0.312237	0.264665
8-25 3L4T	4.2667	14.3	0.313207	0.264665
8-25 3L4T	4.3333	14.3	0.312561	0.264665
8-25 3L4T	4.3333	14.3	0.312561	0.264665
8-25 3L4T	4.3333	14.3	0.312561	0.264665
8-25 3L5T	0.7667	16.6	0.345215	0.299831
8-25 3L5T	0.7	16.6	0.34579	0.299831

8-25 3L5T	0.7	16.6	0.34579	0.299831
8-25 3L5T	0.8	16.6	0.344928	0.299831
8-25 3L5T	0.7	16.6	0.34579	0.299831

Table of Field Test Values

		Irrigation Applied (in)			
		0	1/4	3/4	1 1/2
	a	16.35	16.75	16.675	12.15
	b	20.05	18.375	18.625	3.65
	c	21.4	21.9	20	14.5
	d	15.9	17.2	17.05	6.95
	e	17.9	18.9	17.6	16.4
	f	19.25	19.675	20.15	3.45
	g	16.675	15.5	16.65	11.7
Tag	h	19.8	19.85	19.6	0
ID	i	19.65	18	17.65	15.05
	j	17.1	17.2	16.5	12.15
	k	17.45	17.45	15.85	12.8
	l	20.4	20.9	19.35	14.95
	m	21.1	21.5	20.25	14.55
	n	17.8	18.95	17.65	13.05
	y	19	20.2	19.25	14.85
	z	18.05	18.95	15.35	11.95

Full Resolution Graphs

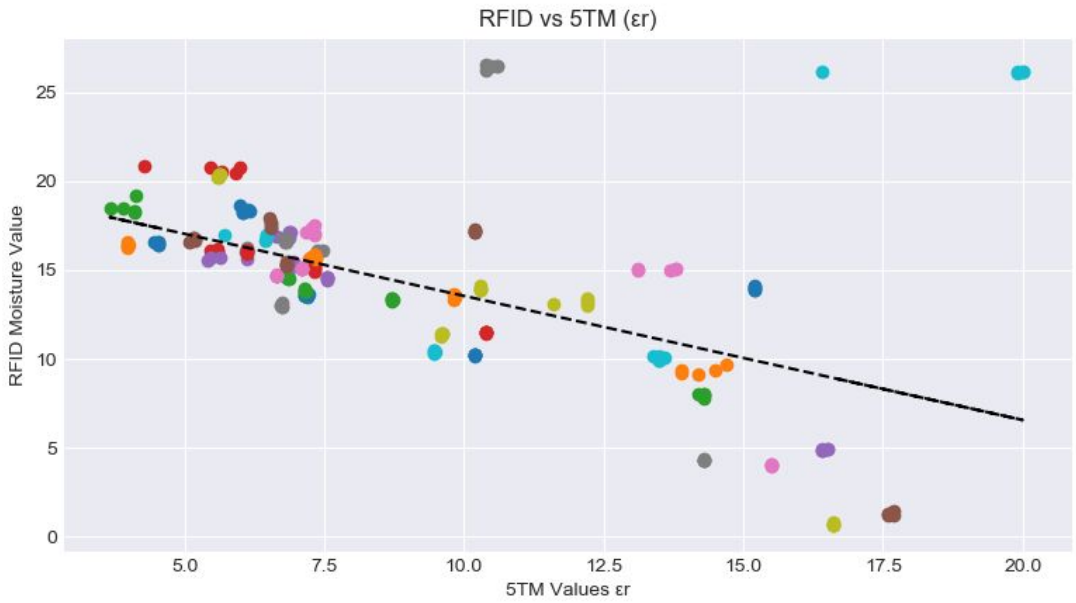

Fig. RFID moisture values vs \varepsilon_r measured by a Meter 5TM soil moisture probe. Outliers were omitted when calculating the regression line.

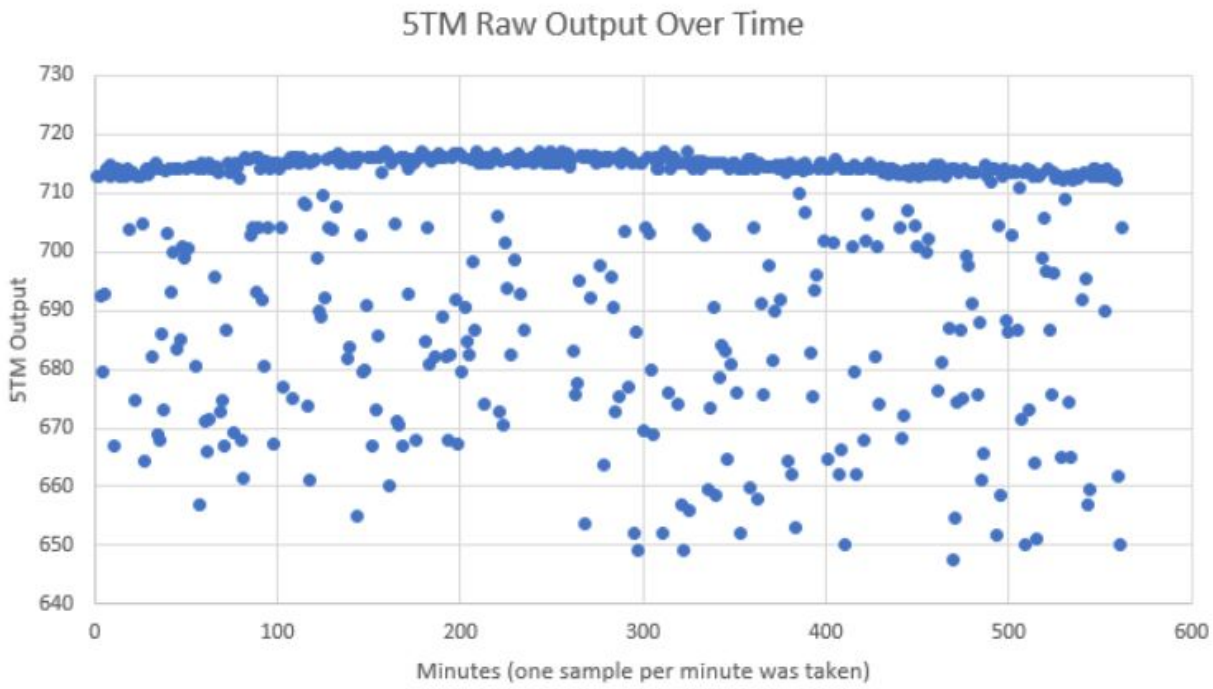

Fig. Graph illustrating left-skew distribution in consecutive 5TM probe readings in an undisturbed soil sample.

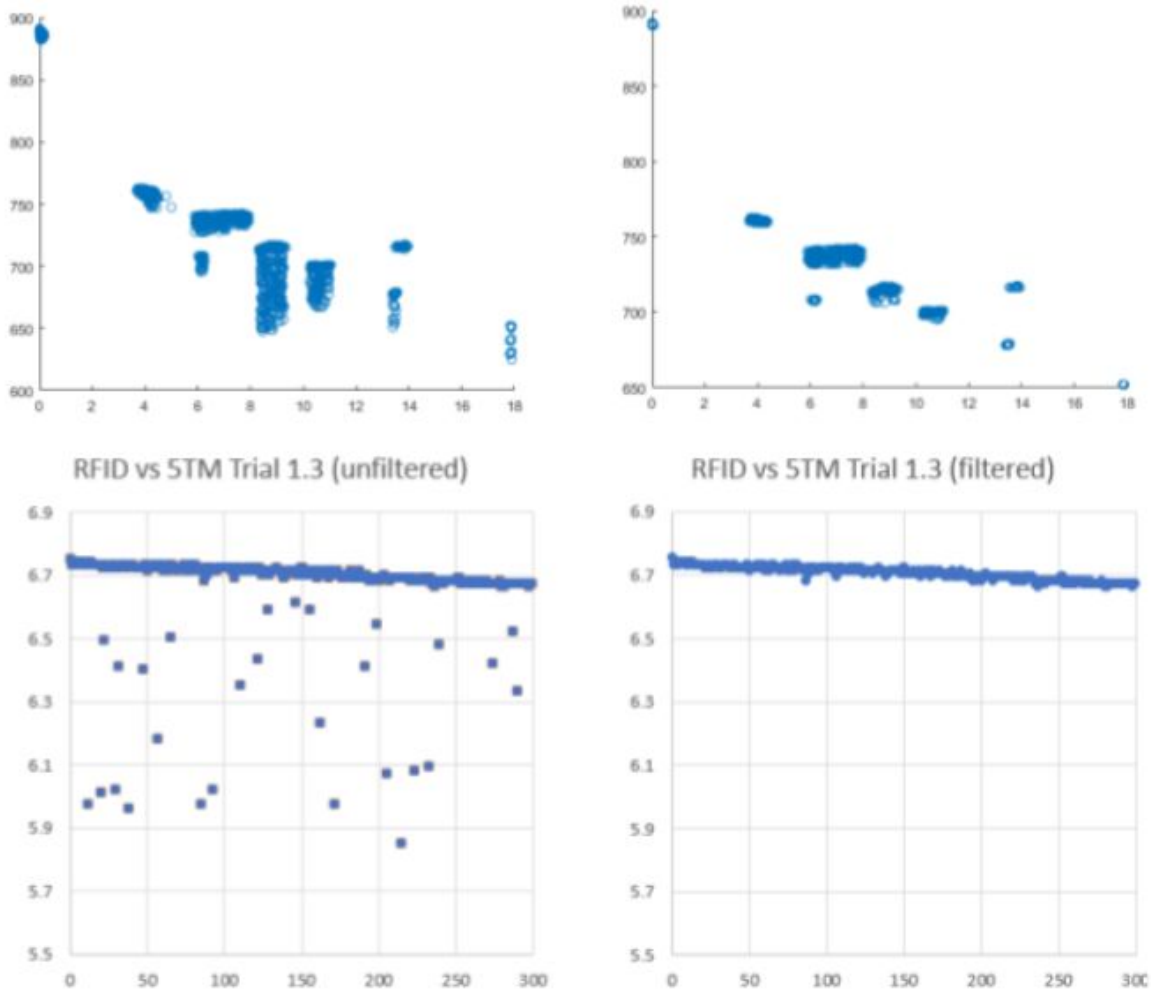

Fig. Illustrations of the effect of proposed filter. Unfiltered (left) vs filtered data (right) for a sample of effected trials (above) and a single zoomed-in trial (below).

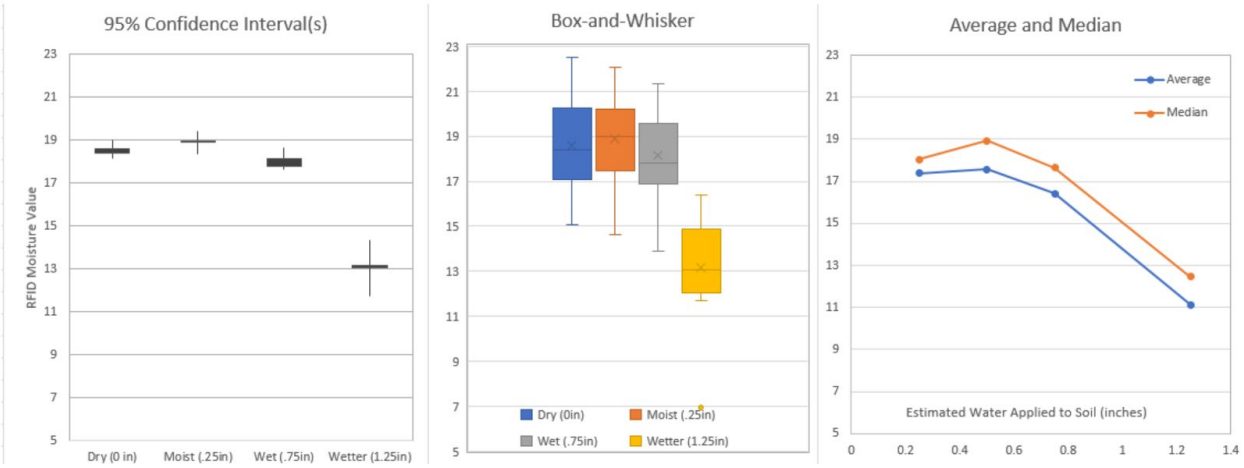

Fig. Data from the field trials visualized.

Plots and Figures Not Included in Thesis

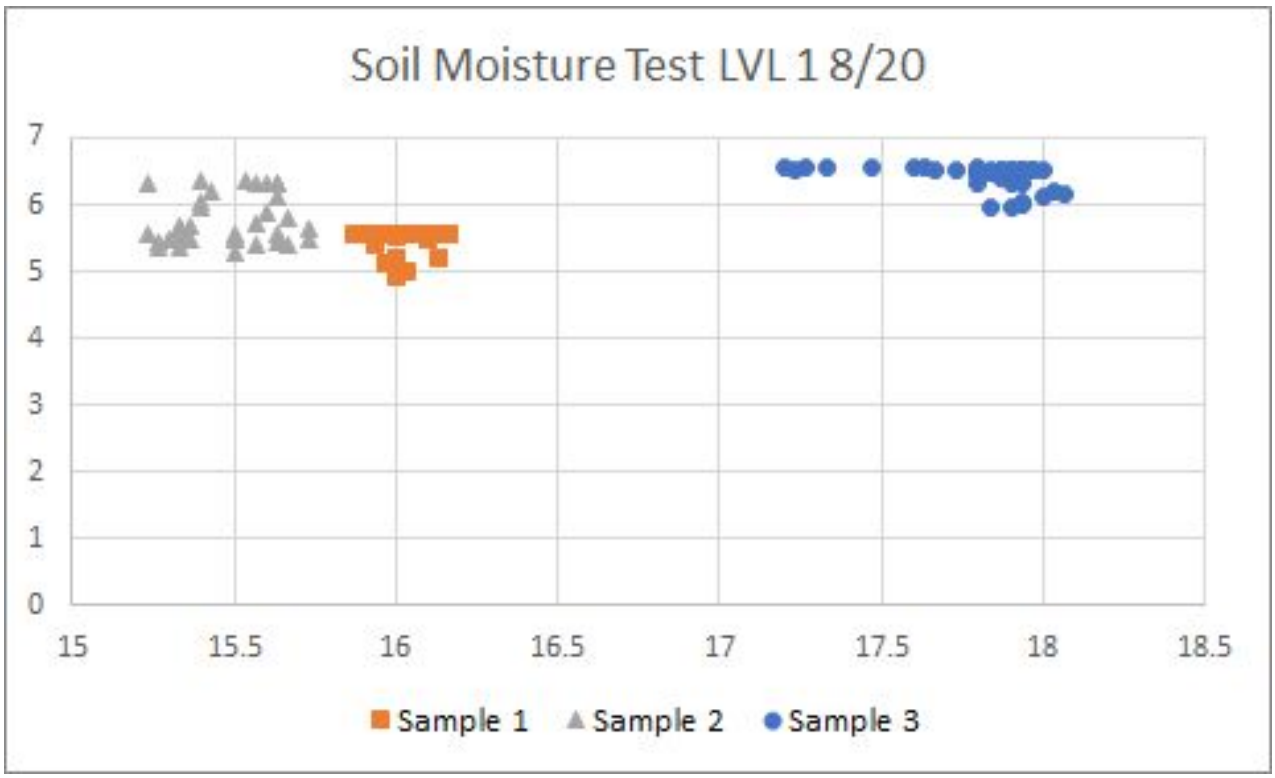

Fig. Sample 3 illustrates a potential outlier caused by an irregularity in the soil. Both from the unnatural shape of the data and how far from the others it is. All of these samples were taken at the same moisture level.

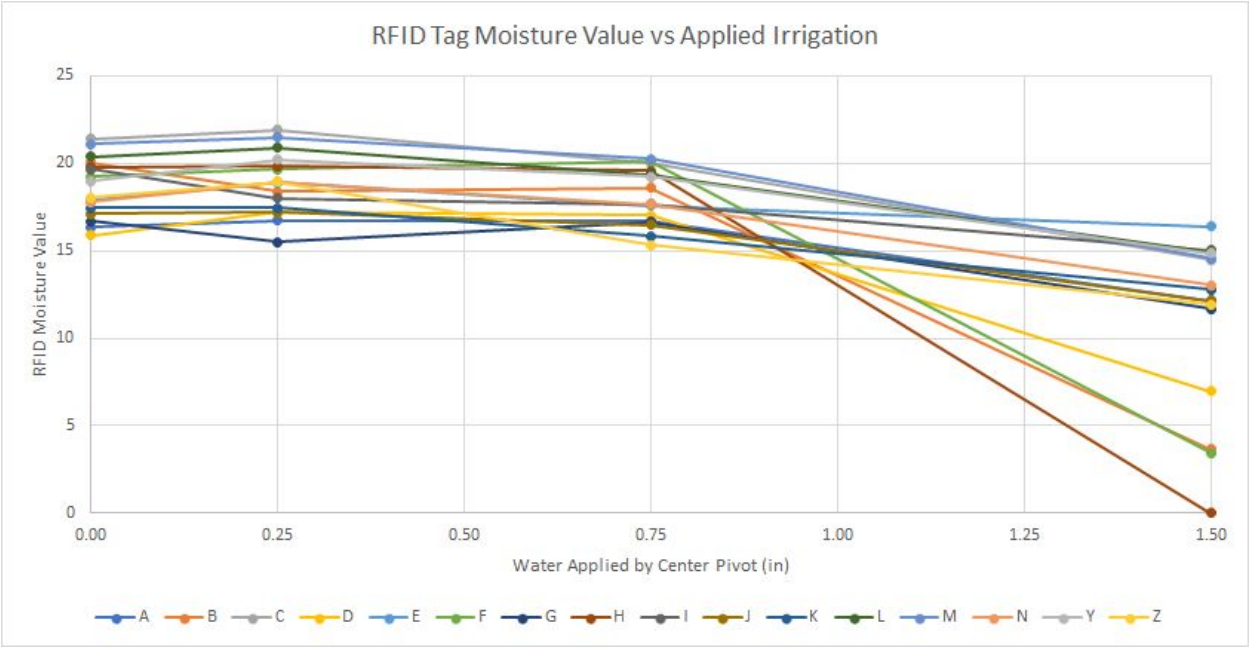

Fig. All tag values for the center pivot test.

Code and Scripts

```
function [ C ] = removeSkew( A_t )
%   Removes outliers from 5TM column in A_t table
%   Returns a new matrix with these data points removed as C
      A = table2array( A_t );
      last = median(A(:,2),2);
      sep = last-mean(A(:,2),2);
      B = [0 0];
      size_ = 1;

      for i=1:(size(A,1))
        current = A(i,2);
        if ( last < (current+ (size(A,1)+5) ) )
            temp = cat(1, B, A(i,1:2));
            B = temp;
            size_ = size_ + 1;
        end
      end
      C = B(2:size_,1:2);
end
```

PROGRAM 1. Matlab filter code using distance from median

```
//Read sensor information
//Caller must provide an array for EPC to be stored in
uint8_t RFID::readTagSensor402(uint8_t *epc, uint8_t &epcLength, uint16_t timeOut)
{
 uint8_t bank = 0x00; //Reserved data bank
 uint8_t address = 0x0B; //Word Address

 return (readData(bank, address, epc, epcLength, timeOut));
}
```

PROGRAM 2. Arduino method added onto Sparkfun RFID library to read RFID soil moisture values from Smartrac's Dogbone RFID tag.

```
#include < SoftwareSerial.h > //Used for transmitting to the device

 SoftwareSerial softSerial(2, 3); //RX, TX
```

```
#include "SparkFun_UHF_RFID_Reader.h" //Library for controlling the M6E Nano module
RFID nano; //Create instance

//custom global variables
#
define NUM_READS 10# define NUM_TRYS_BEFORE_ABORT 8
float RFID_moisture_value;

void setup() {
 Serial.begin(115200);

 //while (!Serial);
 Serial.println();
 Serial.println("Initializing...");

 if (setupRFID(38400) == false) //Configure nano to run at 38400bps
 {
  Serial.println("Module failed to respond. Please check wiring.");
  while (1); //Freeze!
 }

 nano.setRegion(REGION_NORTHAMERICA); //Set to North America

 //TODO
 nano.setReadPower(1500); //15.00 dBm. Higher values may cause USB port to brown out
 //Max Read TX Power is 27.00 dBm and may cause temperature-limit throttling
}

void loop() {
 delay(2000); //2 second delay
 Serial.print("Moisture value: ");
 Serial.println(RFID_moisture_value);
}

boolean RFID_found() {
 byte myEPC[4]; //Most EPCs are 12 bytes
 byte myEPClength;
 byte responseType = 0;

 int counter = 0;
 int moisture_total = 0;

 while (counter < NUM_READS) {
  int exit_pass = 0;
  while (responseType != RESPONSE_SUCCESS) //RESPONSE_IS_TAGFOUND)
  {
```

```
    myEPClength = sizeof(myEPC); //Length of EPC is modified each time .readTagEPC is called

    responseType = nano.readTagSensor402(myEPC, myEPClength, 500); //Scan for a new tag up to
500ms
    //Serial.println(F("Searching for tag"));
    exit_pass++;
    if (exit_pass > NUM_TRYS_BEFORE_ABORT) {
      Serial.println("No tag found");
      RFID_moisture_value = 404; //no tag found value
      return false;
    }
   }
   Serial.println("Found tag");
   moisture_total += tag_to_int(myEPC);
   counter++;
  }
  RFID_moisture_value = ((float) moisture_total) / counter;
  return true;
}

int tag_to_int(byte * tag) {
  //Print EPC
  Serial.print(F(" epc["));
  for (byte x = 0; x < 4; x++) {
   if (tag[x] < 0x10) Serial.print(F("0"));
   Serial.print(tag[x], HEX);
   Serial.print(F(" "));
  }
  Serial.println(F("]"));

  int temp_val = 0;
  temp_val += (int) tag[3]; //least significant bit
  temp_val += ((int) tag[2]) * 16; //most sig bit
  return temp_val;
}

//Gracefully handles a reader that is already configured and already reading continuously
//Because Stream does not have a .begin() we have to do this outside the library
boolean setupRFID(long baudRate) {
  nano.begin(softSerial); //Tell the library to communicate over software serial port

  //Test to see if we are already connected to a module
  //This would be the case if the Arduino has been reprogrammed and the module has stayed
powered
  softSerial.begin(baudRate); //For this test, assume module is already at our desired baud rate
  while (!softSerial); //Wait for port to open
```

```
  //About 200ms from power on the module will send its firmware version at 115200. We need to
ignore this.
  while (softSerial.available()) softSerial.read();

  nano.getVersion();

  if (nano.msg[0] == ERROR_WRONG_OPCODE_RESPONSE) {
    //This happens if the baud rate is correct but the module is doing a ccontinuous read
    nano.stopReading();

    Serial.println(F("Module continuously reading. Asking it to stop..."));

    delay(1500);
  } else {
    //The module did not respond so assume it's just been powered on and communicating at
115200bps
    softSerial.begin(115200); //Start software serial at 115200

    nano.setBaud(baudRate); //Tell the module to go to the chosen baud rate. Ignore the response msg

    softSerial.begin(baudRate); //Start the software serial port, this time at user's chosen baud rate
  }

  //Test the connection
  nano.getVersion();
  if (nano.msg[0] != ALL_GOOD) return (false); //Something is not right

  //The M6E has these settings no matter what
  nano.setTagProtocol(); //Set protocol to GEN2

  nano.setAntennaPort(); //Set TX/RX antenna ports to 1

  return (true); //We are ready to rock
}
```

PROGRAM 3. Example implementation using the modified library.

```
/*
  Library for controlling the Nano M6E from ThingMagic
  This is a stripped down implementation of the Mercury API from ThingMagic
  By: Nathan Seidle @ SparkFun Electronics
  Date: October 3rd, 2016
  https://github.com/sparkfun/Simultaneous_RFID_Tag_Reader
```

```
 Appended By: Brett Stoddard @ Oregon State University Open Source Environmental Sensing Lab
 Date: April 11, 2017
 https://github.com/sparkfun/Simultaneous_RFID_Tag_Reader
 License: Open Source MIT License
 If you use this code please consider buying an awesome board from SparkFun. It's a ton of
 work (and a ton of fun!) to put these libraries together and we want to keep making neat stuff!
 https://opensource.org/licenses/MIT
 The above copyright notice and this permission notice shall be included in all copies or
 substantial portions of the Software.
*/

#include "Arduino.h" //Needed for Stream

#define MAX_MSG_SIZE 255

#define TMR_SR_OPCODE_VERSION 0x03
#define TMR_SR_OPCODE_SET_BAUD_RATE 0x06
#define TMR_SR_OPCODE_READ_TAG_ID_SINGLE 0x21
#define TMR_SR_OPCODE_READ_TAG_ID_MULTIPLE 0x22
#define TMR_SR_OPCODE_WRITE_TAG_ID 0x23
#define TMR_SR_OPCODE_WRITE_TAG_DATA 0x24
#define TMR_SR_OPCODE_KILL_TAG 0x26
#define TMR_SR_OPCODE_READ_TAG_DATA 0x28
#define TMR_SR_OPCODE_CLEAR_TAG_ID_BUFFER 0x2A
#define TMR_SR_OPCODE_MULTI_PROTOCOL_TAG_OP 0x2F
#define TMR_SR_OPCODE_GET_READ_TX_POWER 0x62
#define TMR_SR_OPCODE_GET_WRITE_TX_POWER 0x64
#define TMR_SR_OPCODE_GET_POWER_MODE 0x68
#define TMR_SR_OPCODE_GET_READER_OPTIONAL_PARAMS 0x6A
#define TMR_SR_OPCODE_GET_PROTOCOL_PARAM 0x6B
#define TMR_SR_OPCODE_SET_ANTENNA_PORT 0x91
#define TMR_SR_OPCODE_SET_TAG_PROTOCOL 0x93
#define TMR_SR_OPCODE_SET_READ_TX_POWER 0x92
#define TMR_SR_OPCODE_SET_WRITE_TX_POWER 0x94
#define TMR_SR_OPCODE_SET_REGION 0x97
#define TMR_SR_OPCODE_SET_READER_OPTIONAL_PARAMS 0x9A
#define TMR_SR_OPCODE_SET_PROTOCOL_PARAM 0x9B

#define COMMAND_TIME_OUT  2000 //Number of ms before stop waiting for response from module

//Define all the ways functions can return
#define ALL_GOOD                    0
#define ERROR_COMMAND_RESPONSE_TIMEOUT  1
#define ERROR_CORRUPT_RESPONSE         2
#define ERROR_WRONG_OPCODE_RESPONSE     3
#define ERROR_UNKNOWN_OPCODE           4
#define RESPONSE_IS_TEMPERATURE        5
```

```
#define RESPONSE_IS_KEEPALIVE         6
#define RESPONSE_IS_TEMPTHROTTLE      7
#define RESPONSE_IS_TAGFOUND          8
#define RESPONSE_IS_NOTAGFOUND        9
#define RESPONSE_IS_UNKNOWN           10
#define RESPONSE_SUCCESS                    11
#define RESPONSE_FAIL                       12

//Define the allowed regions - these set the internal freq of the module
#define REGION_INDIA       0x04
#define REGION_JAPAN       0x05
#define REGION_CHINA       0x06
#define REGION_EUROPE      0x08
#define REGION_KOREA       0x09
#define REGION_AUSTRALIA   0x0B
#define REGION_NEWZEALAND  0x0C
#define REGION_NORTHAMERICA 0x0D
#define REGION_OPEN        0xFF

class RFID
{
  public:
    RFID(void);

    bool begin(Stream &serialPort = Serial); //If user doesn't specify then Serial will be used

        void enableDebugging(Stream &debugPort = Serial); //Turn on command sending and
response printing. If user doesn't specify then Serial will be used
        void disableDebugging(void);

    void setBaud(long baudRate);
    void getVersion(void);
    void setReadPower(int16_t powerSetting);
    void getReadPower();
    void setWritePower(int16_t powerSetting);
    void getWritePower();
    void setRegion(uint8_t region);
    void setAntennaPort();
    void setAntennaSearchList();
        void setTagProtocol(uint8_t protocol = 0x05);

    void startReading(void); //Disable filtering and start reading continuously
    void stopReading(void); //Stops continuous read. Give 1000 to 2000ms for the module to stop
reading.

    void enableReadFilter(void);
    void disableReadFilter(void);
```

```
  void setReaderConfiguration(uint8_t option1, uint8_t option2);
  void getOptionalParameters(uint8_t option1, uint8_t option2);
  void setProtocolParameters(void);
  void getProtocolParameters(uint8_t option1, uint8_t option2);

  uint8_t parseResponse(void);

  uint8_t getTagEPCBytes(void); //Pull number of EPC data bytes from record response.
  uint8_t getTagDataBytes(void); //Pull number of tag data bytes from record response. Often zero.
  uint16_t getTagTimestamp(void); //Pull timestamp value from full record response
  uint32_t getTagFreq(void); //Pull Freq value from full record response
  int8_t getTagRSSI(void); //Pull RSSI value from full record response

  bool check(void);

  uint8_t readTagEPC(uint8_t *epc, uint8_t &epcLength, uint16_t timeOut = COMMAND_TIME_OUT);
  uint8_t writeTagEPC(char *newID, uint8_t newIDLength, uint16_t timeOut =
COMMAND_TIME_OUT);

  uint8_t readData(uint8_t bank, uint32_t address, uint8_t *dataRead, uint8_t &dataLengthRead,
uint16_t timeOut = COMMAND_TIME_OUT);
        uint8_t writeData(uint8_t bank, uint32_t address, uint8_t *dataToRecord, uint8_t
dataLengthToRecord, uint16_t timeOut = COMMAND_TIME_OUT);

  uint8_t readUserData(uint8_t *userData, uint8_t &userDataLength, uint16_t timeOut =
COMMAND_TIME_OUT);
  uint8_t writeUserData(uint8_t *userData, uint8_t userDataLength, uint16_t timeOut =
COMMAND_TIME_OUT);

        uint8_t readKillPW(uint8_t *password, uint8_t &passwordLength, uint16_t timeOut =
COMMAND_TIME_OUT);
        uint8_t writeKillPW(uint8_t *password, uint8_t passwordLength, uint16_t timeOut =
COMMAND_TIME_OUT);

        uint8_t readAccessPW(uint8_t *password, uint8_t &passwordLength, uint16_t timeOut =
COMMAND_TIME_OUT);
        uint8_t writeAccessPW(uint8_t *password, uint8_t passwordLength, uint16_t timeOut =
COMMAND_TIME_OUT);

        uint8_t readTID(uint8_t *tid, uint8_t &tidLength, uint16_t timeOut = COMMAND_TIME_OUT);

  uint8_t killTag(uint8_t *password, uint8_t passwordLength, uint16_t timeOut =
COMMAND_TIME_OUT);

  void sendMessage(uint8_t opcode, uint8_t *data = 0, uint8_t size = 0, uint16_t timeOut =
COMMAND_TIME_OUT, boolean waitForResponse = true);
```

```
   void sendCommand(uint16_t timeOut = COMMAND_TIME_OUT, boolean waitForResponse = true);

   void printMessageArray(void);

   uint16_t calculateCRC(uint8_t *u8Buf, uint8_t len);

   //Variables

   //This is our universal msg array, used for all communication
   //Before sending a command to the module we will write our command and CRC into it
   //And before returning, response will be recorded into the msg array. Default is 255 bytes.
   uint8_t msg[MAX_MSG_SIZE];

   //uint16_t tags[MAX_NUMBER_OF_TAGS][12]; //Assumes EPC won't be longer than 12 bytes
   //uint16_t tagRSSI[MAX_NUMBER_OF_TAGS];
   //uint16_t uniqueTags = 0;

/*
   Any public functions beyond this line was written by Brett Stoddard of Oregon State University's
Open Source Environmental Lab
*/

   uint8_t readTagSensor401(uint8_t *epc, uint8_t &epcLength, uint16_t timeOut =
COMMAND_TIME_OUT);
   uint8_t readTagSensor402(uint8_t *epc, uint8_t &epcLength, uint16_t timeOut =
COMMAND_TIME_OUT);
   uint8_t readTagSensor403(uint8_t *epc, uint8_t &epcLength, uint16_t timeOut =
COMMAND_TIME_OUT);


  private:

   Stream *_nanoSerial; //The generic connection to user's chosen serial hardware

   Stream *_debugSerial; //The stream to send debug messages to if enabled

   uint8_t _head = 0; //Tracks the length of the incoming message as we poll the software serial

   boolean _printDebug = false; //Flag to print the serial commands we are sending to the Serial port
for debug
};
```

PROGRAM 4. Modified Sparkfun_Simultaneous_RFID_Tag_Reader_Library's implementation file.

```
/*
  Library for controlling the Nano M6E from ThingMagic
  This is a stripped down implementation of the Mercury API from ThingMagic
  By: Nathan Seidle @ SparkFun Electronics
  Date: October 3rd, 2016
  https://github.com/sparkfun/Simultaneous_RFID_Tag_Reader

  Appended By: Brett Stoddard @ Oregon State University Open Source Environmental Sensing Lab
  Date: April 11, 2017
  https://github.com/sparkfun/Simultaneous_RFID_Tag_Reader
  License: Open Source MIT License
  If you use this code please consider buying an awesome board from SparkFun. It's a ton of
  work (and a ton of fun!) to put these libraries together and we want to keep making neat stuff!
  https://opensource.org/licenses/MIT
  The above copyright notice and this permission notice shall be included in all copies or
  substantial portions of the Software.
*/

#include "Arduino.h" //Needed for Stream

#define MAX_MSG_SIZE 255

#define TMR_SR_OPCODE_VERSION 0x03
#define TMR_SR_OPCODE_SET_BAUD_RATE 0x06
#define TMR_SR_OPCODE_READ_TAG_ID_SINGLE 0x21
#define TMR_SR_OPCODE_READ_TAG_ID_MULTIPLE 0x22
#define TMR_SR_OPCODE_WRITE_TAG_ID 0x23
#define TMR_SR_OPCODE_WRITE_TAG_DATA 0x24
#define TMR_SR_OPCODE_KILL_TAG 0x26
#define TMR_SR_OPCODE_READ_TAG_DATA 0x28
#define TMR_SR_OPCODE_CLEAR_TAG_ID_BUFFER 0x2A
#define TMR_SR_OPCODE_MULTI_PROTOCOL_TAG_OP 0x2F
#define TMR_SR_OPCODE_GET_READ_TX_POWER 0x62
#define TMR_SR_OPCODE_GET_WRITE_TX_POWER 0x64
#define TMR_SR_OPCODE_GET_POWER_MODE 0x68
#define TMR_SR_OPCODE_GET_READER_OPTIONAL_PARAMS 0x6A
#define TMR_SR_OPCODE_GET_PROTOCOL_PARAM 0x6B
#define TMR_SR_OPCODE_SET_ANTENNA_PORT 0x91
#define TMR_SR_OPCODE_SET_TAG_PROTOCOL 0x93
#define TMR_SR_OPCODE_SET_READ_TX_POWER 0x92
#define TMR_SR_OPCODE_SET_WRITE_TX_POWER 0x94
#define TMR_SR_OPCODE_SET_REGION 0x97
#define TMR_SR_OPCODE_SET_READER_OPTIONAL_PARAMS 0x9A
#define TMR_SR_OPCODE_SET_PROTOCOL_PARAM 0x9B
```

```
#define COMMAND_TIME_OUT  2000 //Number of ms before stop waiting for response from module

//Define all the ways functions can return
#define ALL_GOOD                    0
#define ERROR_COMMAND_RESPONSE_TIMEOUT  1
#define ERROR_CORRUPT_RESPONSE         2
#define ERROR_WRONG_OPCODE_RESPONSE     3
#define ERROR_UNKNOWN_OPCODE           4
#define RESPONSE_IS_TEMPERATURE        5
#define RESPONSE_IS_KEEPALIVE          6
#define RESPONSE_IS_TEMPTHROTTLE       7
#define RESPONSE_IS_TAGFOUND           8
#define RESPONSE_IS_NOTAGFOUND         9
#define RESPONSE_IS_UNKNOWN           10
#define RESPONSE_SUCCESS                 11
#define RESPONSE_FAIL                   12

//Define the allowed regions - these set the internal freq of the module
#define REGION_INDIA        0x04
#define REGION_JAPAN        0x05
#define REGION_CHINA        0x06
#define REGION_EUROPE       0x08
#define REGION_KOREA        0x09
#define REGION_AUSTRALIA    0x0B
#define REGION_NEWZEALAND   0x0C
#define REGION_NORTHAMERICA 0x0D
#define REGION_OPEN         0xFF

class RFID
{
  public:
    RFID(void);

    bool begin(Stream &serialPort = Serial); //If user doesn't specify then Serial will be used

          void enableDebugging(Stream &debugPort = Serial); //Turn on command sending and
response printing. If user doesn't specify then Serial will be used
          void disableDebugging(void);

    void setBaud(long baudRate);
    void getVersion(void);
    void setReadPower(int16_t powerSetting);
    void getReadPower();
    void setWritePower(int16_t powerSetting);
    void getWritePower();
    void setRegion(uint8_t region);
    void setAntennaPort();
```

```
  void setAntennaSearchList();
        void setTagProtocol(uint8_t protocol = 0x05);

  void startReading(void); //Disable filtering and start reading continuously
  void stopReading(void); //Stops continuous read. Give 1000 to 2000ms for the module to stop reading.

  void enableReadFilter(void);
  void disableReadFilter(void);

  void setReaderConfiguration(uint8_t option1, uint8_t option2);
  void getOptionalParameters(uint8_t option1, uint8_t option2);
  void setProtocolParameters(void);
  void getProtocolParameters(uint8_t option1, uint8_t option2);

  uint8_t parseResponse(void);

  uint8_t getTagEPCBytes(void); //Pull number of EPC data bytes from record response.
  uint8_t getTagDataBytes(void); //Pull number of tag data bytes from record response. Often zero.
  uint16_t getTagTimestamp(void); //Pull timestamp value from full record response
  uint32_t getTagFreq(void); //Pull Freq value from full record response
  int8_t getTagRSSI(void); //Pull RSSI value from full record response

  bool check(void);

  uint8_t readTagEPC(uint8_t *epc, uint8_t &epcLength, uint16_t timeOut = COMMAND_TIME_OUT);
  uint8_t writeTagEPC(char *newID, uint8_t newIDLength, uint16_t timeOut = COMMAND_TIME_OUT);

  uint8_t readData(uint8_t bank, uint32_t address, uint8_t *dataRead, uint8_t &dataLengthRead, uint16_t timeOut = COMMAND_TIME_OUT);
        uint8_t writeData(uint8_t bank, uint32_t address, uint8_t *dataToRecord, uint8_t dataLengthToRecord, uint16_t timeOut = COMMAND_TIME_OUT);

  uint8_t readUserData(uint8_t *userData, uint8_t &userDataLength, uint16_t timeOut = COMMAND_TIME_OUT);
  uint8_t writeUserData(uint8_t *userData, uint8_t userDataLength, uint16_t timeOut = COMMAND_TIME_OUT);

        uint8_t readKillPW(uint8_t *password, uint8_t &passwordLength, uint16_t timeOut = COMMAND_TIME_OUT);
        uint8_t writeKillPW(uint8_t *password, uint8_t passwordLength, uint16_t timeOut = COMMAND_TIME_OUT);

        uint8_t readAccessPW(uint8_t *password, uint8_t &passwordLength, uint16_t timeOut = COMMAND_TIME_OUT);
        uint8_t writeAccessPW(uint8_t *password, uint8_t passwordLength, uint16_t timeOut =
```

```
COMMAND_TIME_OUT);

        uint8_t readTID(uint8_t *tid, uint8_t &tidLength, uint16_t timeOut = COMMAND_TIME_OUT);

  uint8_t killTag(uint8_t *password, uint8_t passwordLength, uint16_t timeOut =
COMMAND_TIME_OUT);

  void sendMessage(uint8_t opcode, uint8_t *data = 0, uint8_t size = 0, uint16_t timeOut =
COMMAND_TIME_OUT, boolean waitForResponse = true);
  void sendCommand(uint16_t timeOut = COMMAND_TIME_OUT, boolean waitForResponse = true);

  void printMessageArray(void);

  uint16_t calculateCRC(uint8_t *u8Buf, uint8_t len);

  //Variables

  //This is our universal msg array, used for all communication
  //Before sending a command to the module we will write our command and CRC into it
  //And before returning, response will be recorded into the msg array. Default is 255 bytes.
  uint8_t msg[MAX_MSG_SIZE];

  //uint16_t tags[MAX_NUMBER_OF_TAGS][12]; //Assumes EPC won't be longer than 12 bytes
  //uint16_t tagRSSI[MAX_NUMBER_OF_TAGS];
  //uint16_t uniqueTags = 0;

/*
  Any public functions beyond this line was written by Brett Stoddard of Oregon State University's
Open Source Environmental Lab
*/

  uint8_t readTagSensor401(uint8_t *epc, uint8_t &epcLength, uint16_t timeOut =
COMMAND_TIME_OUT);
  uint8_t readTagSensor402(uint8_t *epc, uint8_t &epcLength, uint16_t timeOut =
COMMAND_TIME_OUT);
  uint8_t readTagSensor403(uint8_t *epc, uint8_t &epcLength, uint16_t timeOut =
COMMAND_TIME_OUT);


 private:

  Stream *_nanoSerial; //The generic connection to user's chosen serial hardware

  Stream *_debugSerial; //The stream to send debug messages to if enabled

  uint8_t _head = 0; //Tracks the length of the incoming message as we poll the software serial
```

```
    boolean _printDebug = false; //Flag to print the serial commands we are sending to the Serial port
for debug
};
```

PROGRAM 5. Modified Sparkfun_Simultaneous_RFID_Tag_Reader_Library's header file.